



Qualys Cloud Platform v3.x

API Release Notes

Version 3.3

November 12, 2020 (Updated on November 28, 2020)

Qualys Cloud Suite API gives you many ways to integrate your programs and API calls with Qualys capabilities. You'll find all the details in our user guides, available at the time of release. Just log in to your Qualys account and go to Help > Resources.

What's New

[AM API: Automatic Update for Asset Name](#)

[WAS API: API Support for Catalog functions](#)

[WAS API: Retrieve Retest Status for a Finding](#)

URL to the Qualys API Server

The Qualys API URL you should use for API requests depends on the Qualys platform where your account is located.

[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.

AM API: Automatic Update for Asset Name

API affected	/qps/rest/2.0/search/am/hostasset/
New or Updated APIs	Updated API
DTD or XSD changes	No

A host asset name is derived based on the key attributes (such as DNS, NETBIOS,INSTANCE_ID, IP address). Currently, if an attribute name changed, then it did not automatically reflect in the host asset name. Users had to manually rename such host assets.

To avoid this manual and tedious task, we have now introduced the automatic override feature. If you enable this feature, the asset name is automatically updated across your enterprise, whenever any key attribute (such as DNS, NETBIOS,INSTANCE_ID, IP address) changes. The automatically updated asset name is also reflected in the response of search host asset API.

Note: The automatic override for asset name is an optional feature and is disabled by default. To enable this feature for your subscription, please contact Qualys Support.

Once you enable this feature, the host asset name is automatically updated whenever any of the key attribute changes. You can also disable the automatic override for specific assets by manually renaming the asset name and selecting check box to prevent automatic update of asset name.

IMPORTANT UPDATE: This feature is no longer a part of this release and will be available with Qualys Platform version 3.3.1.0.

WAS API: API Support for Catalog functions

API affected	/qps/rest/3.0/get/was/catalog/{id} /qps/rest/3.0/count/was/catalog /qps/rest/3.0/search/was/catalog /qps/rest/3.0/update/was/catalog/{id} /qps/rest/3.0/delete/was/catalog/{id} /qps/rest/3.0/updateEntries/was/catalog /qps/rest/3.0/addToSubscription/was/catalog/{id}
New or Updated APIs	New
DTD or XSD changes	New

With this release, we are providing API support for catalog functions that are currently available only from WAS UI. Using these APIs, you can get the details of the catalog entries, find the total number of entries in the catalog, update, delete the catalog entries, add a web application entry to subscription to create a web application and update the entries in the catalog to add the data discovered in the most recent VM scan results within your account.

[Get catalog entry details](#)

[Search for a catalog entry](#)

[Count catalog entries](#)

[Update catalog entry](#)

[Delete catalog entry](#)

[Update entries in catalog](#)

[Add to subscription](#)

Get catalog entry details

/qps/rest/3.0/get/was/catalog/{id}

[GET]

View the details of a catalog entry that is in your scope. In the output, "Comment" tag will show the comment added by the system and comment added by you.

Permissions

- You must have the WAS module enabled.
- You must have the "API access" and "Access WAS module" permissions.
- You must have the "WAS.CATALOG.ACCESS" permission.

Input Parameters

The element "id" (integer) is required, where "id" identifies the catalog entry.

Sample - View details of a catalog entry

API Request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X  
"https://qualysapi.qualys.com/qps/rest/3.0/get/was/catalog/306904"
```

XML Output

```
<?xml version="1.0" encoding="UTF-8"?>  
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/3.0/w  
as/catalog.xsd">  
  <responseCode>SUCCESS</responseCode>  
  <count>1</count>  
  <data>  
    <Catalog>  
      <id>306904</id>  
      <ipAddress>10.113.196.17</ipAddress>  
      <port>80</port>  
      <operatingSystem>MacOS X 9.0.0</operatingSystem>  
      <source>VM_SCAN</source>  
      <fqdn>10-113-196-17.bogus.tld</fqdn>  
      <netbiosName>SYS_10_113_196_17</netbiosName>  
      <status>NEW</status>  
      <comments>  
        <count>4</count>  
        <list>  
          <Comment>  
            <contents>
```

```

                                <![CDATA[Web Application added from scan
consolidated data from VM]]>
                                </contents>
                                <createdDate>2018-05-18T10:33:55Z</createdDate>
                                </Comment>
                                <Comment>
                                    <contents>
                                        <![CDATA[asdasd]]>
                                    </contents>
                                    <author>
                                        <id>1918433</id>
                                        <username>qualys_joe</username>
                                        <firstName>
                                            <![CDATA[qualys]]>
                                        </firstName>
                                        <lastName>
                                            <![CDATA[joe]]>
                                        </lastName>
                                    </author>
                                    <createdDate>2020-10-22T07:47:25Z</createdDate>
                                </Comment>
                                <Comment>
                                    <contents>
                                        <![CDATA[Entry added to subscription as 'Catalog
Web Application: 10-113-196-17.bogus.tld, Port 80']]>
                                    </contents>
                                    <createdDate>2020-10-12T10:16:45Z</createdDate>
                                </Comment>
                            </list>
                        </comments>
                        <createdDate>2018-05-18T10:33:55Z</createdDate>
                        <updatedAt>2020-10-22T07:47:25Z</updatedAt>
                        <updatedBy>
                            <id>1918433</id>
                            <username>qualys_joe</username>
                            <firstName>
                                <![CDATA[qualys]]>
                            </firstName>
                            <lastName>
                                <![CDATA[joe]]>
                            </lastName>
                        </updatedBy>
                    </Catalog>
                </data>
            </ServiceResponse>

```

Search for a catalog entry

qps/rest/3.0/search/was/catalog

[POST]

Returns a list of catalog entries based on the search criteria.

Permissions

- You must have the WAS module enabled.
- You must have the "API access" and "Access WAS module" permissions.
- You must have the "WAS.CATALOG.ACCESS" permission.

Input Parameters

These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND.

Parameter	Description
id	(integer) The ID of the catalog entry.
ipAddress	(integer) The IP address of the discovered host. We support wild card character * for numbers in IP Address. For example, 10.11.196.* or 10.11.*.* are valid patterns for IP address. Examples of Invalid patterns: *1.123.123.123, 1*1.123.123.123 and 1*.123.123.123
port	(integer) The port number of the discovered service.
source	(text) The source of the catalog entries. Valid values are: VM_SCAN, VM_MAP, and WAS_SCAN.
Status	(text) The status of the entry. Valid values are NEW, ROGUE, APPROVED, IGNORED, IN_SUBSCRIPTION.
operatingSystem	(text) The operating system of discovered host.
netbiosName	(text) The NetBIOS name of the discovered host.
fqdn	(text) The fully qualified domain name of the discovered host.
createdDate	(date) The date and time when the catalog entry is created. The date format is YYYY-MM-DDTHH:MM:SSZ. For example: 2018-05-18T10:33:54Z
UpdatedDate	(date) The updated date and time when the catalog entry is updated. The date format is YYYY-MM-DDTHH:MM:SSZ. For example: 2018-05-18T10:33:54Z.

Sample - Search catalog entries with IP address

Let us view all catalog entries in the user's scope.

API Request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --  
data-binary @-  
"https://qualysapi.qualys.com/qps/rest/3.0/search/was/catalog" <  
file.xml  
Note: "file.xml" contains the request POST data.
```

Request POST data

```
<ServiceRequest>  
  <filters>  
    <Criteria field="ipAddress"  
      operator="EQUALS">10.113.*.*</Criteria>  
  </filters>  
</ServiceRequest>
```

XML Output

```
<?xml version="1.0" encoding="UTF-8"?>  
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
  xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/3.0/w  
as/catalog.xsd">  
  <responseCode>SUCCESS</responseCode>  
  <count>2</count>  
  <hasMoreRecords>false</hasMoreRecords>  
  <data>  
    <Catalog>  
      <id>306909</id>  
      <ipAddress>10.113.196.192</ipAddress>  
      <port>443</port>  
      <operatingSystem>Ubuntu / Fedora / Tiny Core Linux / Linux  
        3.x</operatingSystem>  
      <source>VM_SCAN</source>  
      <status>ROGUE</status>  
      <createdDate>2018-05-18T10:33:55Z</createdDate>  
      <updatedDate>2020-05-19T13:50:08Z</updatedDate>  
      <updatedBy>  
        <id>1918433</id>  
        <username>qualys_joe</username>  
        <firstName>  
          <![CDATA[qualys]]>  
        </firstName>  
        <lastName>  
          <![CDATA[joe]]>  
        </lastName>
```

```

        </updatedBy>
    </Catalog>
    <Catalog>
        <id>306906</id>
        <ipAddress>10.113.196.18</ipAddress>
        <port>80</port>
        <operatingSystem>Windows XP Service Pack 2</operatingSystem>
        <source>VM_SCAN</source>
        <fqdn>10-113-196-18.bogus.tld</fqdn>
        <netbiosName>SYS_10_113_196_18</netbiosName>
        <status>ROGUE</status>
        <createdDate>2018-05-18T10:33:55Z</createdDate>
        <updatedAt>2020-05-19T13:50:08Z</updatedAt>
        <updatedBy>
            <id>1918433</id>
            <username>qualys_joe</username>
            <firstName>
                <![CDATA[qualys]]>
            </firstName>
            <lastName>
                <![CDATA[joe]]>
            </lastName>
        </updatedBy>
    </Catalog>
</data>
</ServiceResponse>

```

Count catalog entries

/qps/rest/3.0/count/was/catalog

[GET] [POST]

Returns the total number of catalog entries in the user's scope.

Permissions

- You must have the WAS module enabled.
- You must have "API access" and "Access WAS module" permissions.
- You must have the "WAS.CATALOG.ACCESS" permission.

Input Parameters

These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND.

Parameter	Description
id	(integer) The ID of the catalog entry.

Parameter	Description
ipAddress	(integer) The IP address of the discovered host. We support wild card character * for numbers in IP Address. For example, 10.11.196.* or 10.11.*.* are valid patterns for IP address. Examples of Invalid patterns: *1.123.123.123, 1*1.123.123.123 and 1*.123.123.123
port	(integer) The port number of the discovered service.
source	(text) The source of the catalog entries. Valid values are: VM_SCAN, VM_MAP, and WAS_SCAN.
Status	(text) The status of the entry. Valid values are NEW, ROGUE, APPROVED, IGNORED, IN_SUBSCRIPTION.
operatingSystem	(text) The operating system of discovered host.
netbiosName	(text) The NetBIOS name of the discovered host.
fqdn	(text) The fully qualified domain name of the discovered host.
createdDate	(date) The date and time when the catalog entry is created. The date format is YYYY-MM-DDTHH:MM:SSZ. For example: 2018-05-18T10:33:54Z
UpdatedDate	(date) The updated date and time when the catalog entry is updated. The date format is YYYY-MM-DDTHH:MM:SSZ. For example: 2018-05-18T10:33:54Z.

Sample - Get count of catalog entries (no criteria)

Returns the number (count) of all catalog entries in the user's scope.

API Request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X
"https://qualysapi.qualys.com/qps/rest/3.0/count/was/catalog"
```

XML Output

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/3.0/w
as/catalog.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1355</count>
</ServiceResponse>
```

Update catalog entry

qps/rest/3.0/update/was/catalog/{id}

[POST]

Updates the status and comments for a catalog entry which is in your scope. Want to find an ID of a catalog entry to use as input? See Search catalog entries.

Permissions

- You must have the WAS module enabled.
- You must have the "API access" and "Access WAS module" permissions.
- You must have the "WAS.CATALOG.ACCESS" and "WAS.CATALOG.ENTRY.UPDATE" permissions.

Input Parameters

Parameter	Description
id	(integer) The element "id" is required, where "id" identifies a catalog entry.
status	(text) This is an optional parameter. The status can be updated to one of these statuses: ROGUE, NEW, APPROVED and IGNORED. IN_SUBSCRIPTION status can not be updated using the Update API.
Comments	(text) This is an optional parameter. You can add comments but you can not update/delete existing comments.

Sample - Update a catalog entry

Let us update the status of the catalog entry and add a comment to the entry.

API Request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --  
data-binary @-  
"https://qualysapi.qualys.com/qps/rest/3.0/update/was/catalog/368106" <  
file.xml
```

Note: "file.xml" contains the request POST data.

Request POST data

```
<ServiceRequest>  
  <data>  
    <Catalog>  
      <status>ROGUE</status>  
      <comments>
```

```
        <add>
          <Comment>
            <contents>
              <![CDATA[Comment 1]]>
            </contents>
          </Comment>
        </add>
      </comments>
    </Catalog>
  </data>
</ServiceRequest>
```

XML Output

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/3.0/w
as/catalog.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <Catalog>
      <id>368106</id>
    </Catalog>
  </data>
</ServiceResponse>
```

Delete catalog entry

/qps/rest/3.0/delete/was/catalog/{id}

[POST]

Deletes a catalog entry which is in your scope. Want to find an ID of a catalog entry to use as input? See Search catalog entries.

Permissions

- You must have the WAS module enabled.
- You must have the "API access" and "Access WAS module" permissions.
- You must have the "WAS.CATALOG.ACCESS" and "WAS.CATALOG.ENTRY.DELETE" permissions.

Input Parameters

The element “id” (integer) is required, where “id” identifies a catalog entry.

These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND.

Parameter	Description
id	(integer) The element "id" identifies a catalog entry.
ipAddress	(integer) The IP address of the discovered host. We support wild card character * for numbers in IP Address. For example, 10.11.196.* or 10.11.*.* are valid patterns for IP address. Examples of Invalid patterns: *1.123.123.123, 1*1.123.123.123 and 1*.123.123.123
port	(integer) The port number of the discovered service.
source	(text) The source of the catalog entries. Valid values are: VM_SCAN, VM_MAP, and WAS_SCAN.
Status	(text) The status of the entry. Valid values are NEW, ROGUE, APPROVED, IGNORED, IN_SUBSCRIPTION.
operatingSystem	(text) The operating system of discovered host.
netbiosName	(text) The NetBIOS name of the discovered host.
fqdn	(text) The fully qualified domain name of the discovered host.
createdDate	(date) The date and time when the catalog entry is created. The date format is YYYY-MM-DDTHH:MM:SSZ. For example: 2018-05-18T10:33:54Z
UpdatedDate	(date) The updated date and time when the catalog entry is updated. The date format is YYYY-MM-DDTHH:MM:SSZ. For example: 2018-05-18T10:33:54Z.

Sample - Delete a catalog entry

Let us delete a catalog entry with ID 368106.

API Request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
"https://qualysapi.qualys.com/qps/rest/3.0/delete/was/catalog/368106"
```

XML Output

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/3.0/w
as/catalog.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
```

```
<Catalog>
  <id>368106</id>
</Catalog>
</data>
</ServiceResponse>
```

Update entries in catalog

/qps/rest/3.0/updateEntries/was/catalog

[POST]

Updates the entries in the catalog to add data discovered in the most recent VM scan results within your account.

Permissions

- You must have the WAS module enabled.
- You must have the "API access" and "Access WAS module" permissions.
- You must have the "WAS.CATALOG.ACCESS" and "WAS.CATALOG.UPDATE" permissions.

Sample - Update entries in the catalog

API Request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
"https://qualysapi.qualys.com/qps/rest/3.0/updateEntries/was/catalog"
```

XML Output

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/3.0/w
as/catalog.xsd">
  <responseCode>SUCCESS</responseCode>
</ServiceResponse>
```

Add to subscription

/qps/rest/3.0/addToSubscription/was/catalog/{id}

[POST]

Adds a web application entry to subscription to create a web application.

Permissions

- You must have the WAS module enabled.

- You must have the "API access" and "Access WAS module" permissions.
- You must have the "WAS.CATALOG.ACCESS" and "WAS.CATALOG.ENTRY.ADD_TO_SUBSCRIPTION" permissions.

Input Parameters

The element "id" (integer) is required, where "id" identifies the catalog entry.

Sample - Add a catalog entry to subscription

API Request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
"https://qualysapi.qualys.com/qps/rest/3.0/addToSubscription/was/catalog/
306904"
```

XML Output

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/versi
on.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <success>
      <ids>413904,413906</ids>
      <count>1</count>
    </success>
    <duplicate>
      <count>1</count>
      <ids>413905</ids>
    </duplicate>
    <error>
      <count>2</count>
      <errorMessage>Invalid URL for web application catalog
entries: 413907Some error occurred for web application
catalog entries:413908
      </errorMessage>
    </error>
  </data>
</ServiceResponse>
```

New XSD

<platform API server>/qps/xsd/3.0/was/catalog.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
```

```

        elementFormDefault="qualified">

    <!-- REQUEST -->
    <xs:element name="ServiceRequest">
        <xs:complexType>
            <xs:all>
                <xs:element name="filters" type="ServiceRequestFilters"
minOccurs="0"/>
                <xs:element name="preferences"
type="ServiceRequestPreferences" minOccurs="0"/>
                <xs:element name="data" type="ServiceRequestData"
minOccurs="0"/>
            </xs:all>
        </xs:complexType>
    </xs:element>

    <xs:complexType name="ServiceRequestFilters">
        <xs:sequence>
            <xs:element name="Criteria" type="Criteria"
maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>

    <xs:complexType name="Criteria">
        <xs:simpleContent>
            <xs:extension base="xs:string">
                <xs:attribute name="field" type="xs:string"/>
                <xs:attribute name="operator">
                    <xs:simpleType>
                        <xs:restriction base="xs:string">
                            <xs:enumeration value="CONTAINS"/>
                            <xs:enumeration value="IN"/>
                            <xs:enumeration value="EQUALS"/>
                            <xs:enumeration value="NOT EQUALS"/>
                            <xs:enumeration value="GREATER"/>
                            <xs:enumeration value="LESSER"/>
                            <xs:enumeration value="NONE"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:attribute>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>

    <xs:complexType name="ServiceRequestPreferences">
        <xs:all>
            <xs:element name="startFromId" type="xs:long" minOccurs="0"/>
            <xs:element name="startFromOffset" type="xs:int"
minOccurs="0"/>

```

```

        <xs:element name="limitResults" type="xs:int" minOccurs="0"/>
    </xs:all>
</xs:complexType>

<xs:complexType name="ServiceRequestData">
    <xs:sequence>
        <xs:element name="Catalog" type="Catalog" minOccurs="0"/>
    </xs:sequence>
</xs:complexType>

<!-- RESPONSE -->
<xs:element name="ServiceResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="responseCode" type="ResponseCode"/>
            <xs:element name="responseErrorDetails"
type="ResponseErrorObject" minOccurs="0"/>
            <xs:element name="count" type="xs:int" minOccurs="0"/>
            <xs:element name="hasMoreRecords" type="xs:boolean"
minOccurs="0"/>
            <xs:element name="lastId" type="xs:long" minOccurs="0"/>
            <xs:element name="data" type="ServiceResponseData"
minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>

<xs:simpleType name="ResponseCode">
    <xs:restriction base="xs:string">
        <xs:enumeration value="AUTH_CREDENTIALS_NEEDED"/>
        <xs:enumeration value="CANNOT_BE_NULL"/>
        <xs:enumeration value="INVALID_XML"/>
        <xs:enumeration value="INVALID_CREDENTIALS"/>
        <xs:enumeration value="INVALID_API_VERSION"/>
        <xs:enumeration value="INVALID_PARAM"/>
        <xs:enumeration value="INVALID_URL"/>
        <xs:enumeration value="INVALID_REQUEST"/>
        <xs:enumeration value="NOT_FOUND"/>
        <xs:enumeration value="OTHER_ERROR"/>
        <xs:enumeration value="OPERATION_NOT_SUPPORTED"/>
        <xs:enumeration value="EVALUATION_EXPIRED"/>
        <xs:enumeration value="JMS_SERVER_DOWN"/>
        <xs:enumeration value="RMI_SERVER_DOWN"/>
        <xs:enumeration value="SUCCESS"/>
        <xs:enumeration value="STILL_PROCESSING"/>
        <xs:enumeration value="UNAUTHORIZED"/>
        <xs:enumeration value="UNAUTHORIZED_DESTINATION_APPS"/>
    </xs:restriction>
</xs:simpleType>

```

```

        <xs:enumeration value="UNIDENTIFIED_PRODUCER"/>
        <xs:enumeration value="UNKNOWN_OBJECT"/>
    </xs:restriction>
</xs:simpleType>

<xs:complexType name="ResponseErrorObject">
    <xs:sequence>
        <xs:element name="errorMessage" type="xs:string"/>
        <xs:element name="errorResolution" type="xs:string"
minOccurs="0"/>
        <xs:element name="internalErrorCodeId" type="xs:int"
minOccurs="0"/>
    </xs:sequence>
</xs:complexType>

<xs:complexType name="ServiceResponseData">
    <xs:sequence>
        <xs:element name="Catalog" type="Catalog" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>

<xs:complexType name="User">
    <xs:all>
        <xs:element name="id" type="xs:long" minOccurs="0"/>
        <xs:element name="username" type="xs:string" minOccurs="0"/>
        <xs:element name="firstName" type="Cdata" minOccurs="0"/>
        <xs:element name="lastName" type="Cdata" minOccurs="0"/>
    </xs:all>
</xs:complexType>

<xs:complexType name="Cdata">
    <xs:simpleContent>
        <xs:extension base="xs:string"/>
    </xs:simpleContent>
</xs:complexType>

<xs:complexType name="CommentList">
    <xs:all>
        <xs:element name="count" type="xs:int" minOccurs="0"/>
        <xs:element name="list" minOccurs="0" >
            <xs:complexType>
                <xs:sequence>
                    <xs:element name="Comment" type="Comment"
minOccurs="0" maxOccurs="unbounded"/>
                </xs:sequence>
            </xs:complexType>
        </xs:element>
        <xs:element name="set" minOccurs="0" >

```

```

        <xs:complexType>
          <xs:sequence>
            <xs:element name="Comment" type="Comment"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="add" minOccurs="0" >
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Comment" type="Comment"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:all>
  </xs:complexType>

  <xs:complexType name="Comment">
    <xs:all>
      <xs:element name="contents" type="Cdata"/>
      <xs:element name="author" type="User" minOccurs="0"/>
      <xs:element name="createdDate" type="xs:dateTime"
minOccurs="0"/>
    </xs:all>
  </xs:complexType>

  <xs:complexType name="Catalog">
    <xs:sequence>
      <xs:element name="id" type="xs:long" minOccurs="0"/>
      <xs:element name="ipAddress" type="xs:string" minOccurs="0"/>
      <xs:element name="port" type="xs:integer" minOccurs="0"/>
      <xs:element name="operatingSystem" type="xs:string"
minOccurs="0"/>
      <xs:element name="source" type="CatalogSource" minOccurs="0"/>
      <xs:element name="fqdn" type="xs:string" minOccurs="0"/>
      <xs:element name="netbiosName" type="xs:string"
minOccurs="0"/>
      <xs:element name="status" type="CatalogStatus" minOccurs="0"/>
      <xs:element name="comments" type="CommentList" minOccurs="0"/>
      <xs:element name="createdDate" type="xs:dateTime"
minOccurs="0"/>
      <xs:element name="updatedAt" type="xs:dateTime"
minOccurs="0"/>
      <xs:element name="updatedBy" type="User" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>

  <xs:simpleType name="CatalogSource">

```

```
        <xs:restriction base="xs:string">
            <xs:enumeration value="VM_MAP"/>
            <xs:enumeration value="VM_SCAN"/>
            <xs:enumeration value="WAS_SCAN"/>
        </xs:restriction>
    </xs:simpleType>

    <xs:simpleType name="CatalogStatus">
        <xs:restriction base="xs:string">
            <xs:enumeration value="NEW"/>
            <xs:enumeration value="ROGUE"/>
            <xs:enumeration value="APPROVED"/>
            <xs:enumeration value="IGNORED"/>
            <xs:enumeration value="IN_SUBSCRIPTION"/>
        </xs:restriction>
    </xs:simpleType>

</xs:schema>
```

WAS API: Retrieve Retest Status for a Finding

API affected	/qps/rest/3.0/retestStatus/was/finding/{id}
New or Updated APIs	New
DTD or XSD changes	No

We have introduced a new API "Retrieve Retest Status" API to retrieve the status of the retest for a finding. This will help you to automate scanning and retesting processes. The API will return one of these statuses: NO_RETEST, UNDER_RETEST, RETESTED, CANCELING, and CANCELED

Permissions

- You must have the WAS module enabled.
- You must have the "API access" and "Access WAS module" permissions.
- You must have the View permission.

Input Parameters

The API supports POST method. The Input parameters are id or uniqueId. We support optional filters that are available for the Search Finding API.

Parameter	Description
id	(integer) ID of the finding (WebAppVuln or WebAppSensitiveContent).
uniqueId	(value) The 36-bit unique id assigned to the finding.

Sample - Retrieve retest status for a finding

API Request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
"https://qualysapi.qualys.com/qps/rest/3.0/retestStatus/was/finding/2730074"
```

XML Output

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/3.0/was/finding.xsd">
<responseCode>SUCCESS</responseCode>
<count>1</count>
<data>
<Retest>
<retestStatus>UNDER_RETEST</retestStatus>
```

```
</Retest>  
</data>  
</ServiceResponse>
```