



Asset Management and Tagging API v1

User Guide
Version 2.3

February 18, 2014

Copyright 2013-2014 by Qualys, Inc. All Rights Reserved.

Qualys, the Qualys logo and QualysGuard are registered trademarks of Qualys, Inc. All other trademarks are the property of their respective owners.

Qualys, Inc.
1600 Bridge Parkway
Redwood Shores, CA 94065
1 (650) 801 6100

CONTENTS

Preface

Chapter 1 Welcome

Introduction to the API Paradigm	8
Request URL	8
Making Requests with an XML Payload	8
Curl	8
XML Output Pagination / Truncation Logic	9
Base URL to the Qualys API Server	12
Authentication	12

Chapter 2 Tags API

Create Tag	14
Update Tag	18
Get Tag	21
Count Tags	24
Search Tags - By Asset Count	25
Search Tags - By Criteria	27
Delete Tag	30

Chapter 3 Host Asset API

Create Host Asset	32
Update Host Asset	39
Get Host Asset	43
Count Host Assets	48
Search Assigned Host Assets - By ID	49
Search Hosts Assets - By Criteria	51
Delete Host Asset	56

Chapter 4 Asset API

Update Asset	58
Get Asset	61
Count Assets	63
Search Assets - By SOLR Keyword	64
Search Assigned Assets - By ID	66
Search Assets - By Criteria	68
Delete Asset	71

Preface

Using the Qualys Asset Management and Tagging API, third parties can integrate the Qualys Security and Compliance solution into their own applications using an extensible XML interface. This user guide is intended for application developers who will use the Qualys Asset Management and Tagging API.

About Qualys

Qualys, Inc. (NASDAQ: QLYS), is a pioneer and leading provider of cloud security and compliance solutions with over 6,000 customers in more than 100 countries, including a majority of each of the Forbes Global 100 and Fortune 100. The Qualys Cloud Platform and integrated suite of solutions help organizations 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. 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

Welcome

Welcome to Qualys Asset Management and Tagging API. The sections that follow describe how to use the Tags API, Host Asset API and the Asset API.

We recommend you review these topics before you get started:

[Introduction to the API Paradigm](#) - We recommend you review important information about the API framework.

[Base URL to the Qualys API Server](#) - We'll give you the basics about making API requests. The base URL depends on the platform where your Qualys account is located.

[Authentication](#) - We'll tell you about the method used for authentication. API requests must authenticate using Qualys credentials.

Introduction to the API Paradigm

The Qualys Asset Management and Tagging API framework offers numerous innovations and new functionality compared to the other Qualys API frameworks.

Request URL

The URL for making Asset Management and Tagging API requests respects the following structure:

https://<baseurl>/qps/rest/1.0/<operation>/<module>/<object>/<object_id>

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://qualysapi.qualys.com
<operation>	The request operation, such as get a list, get a count, search, create, and update.
<module>	The API module. For the Asset Management and Tagging API, the module is: "am".
<object>	The module specific object.
<object_id>	(Optional) The module specific object ID, if appropriate.

Making Requests with an XML Payload

While it is still possible to create simple API requests using the GET method, you can create API requests using the POST method with an XML payload to make an advanced request.

The XML payloads can be compared to a scripting language that allows user to make multiple actions within one single API request, like adding a parameter to an object and updating another parameter.

The XML structure of the payload is described in the XSD files.

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 Asset Management and Tagging API requests using the HTTP over SSL (https) protocol, which is required by the Qualys API framework.

Want to learn more? Visit <http://curl/haxx/se>

The following Curl options are used according to different situations:

Option	Description
-u "LOGIN:PASSWORD"	This option is used for basic authentication.
-X "POST"	This option is used to provide a method other than the default method, GET.
-H "content-type"	This option is used to provide a custom HTTP request header parameter for content type, to specify the MIME type of the curl's payload.
--data-binary	This option is used to specify the POST data. See the examples below.

The sample below shows a typical Curl request using options mentioned above and additional explanation of how they interact with each other. The option -X "POST" tells Curl to execute the request using the HTTP POST method. The option "--data-binary @" tells Curl to read the POST data from its standard input (stdin). The string "< file.xml" is interpreted by the shell to redirect the content of the file to the stdin of the command. The option -H "content-type: text/xml" tells Curl the POST data in "file.xml" is XML in text format.

```
curl -H "content-type: text/xml" -X "POST" --data-binary @- "https://example.com" < file.xml
```

This documentation uses Curl examples showing the POST data in the "file.xml" file. This is referred to as Request POST Data. This can also be referred to as the Payload.

XML Output Pagination / Truncation Logic

The XML output of a search API request is paginated and the default page size is 100 object records. The page size can be customized to a value between 1 and 1,000. If the number of records is greater than the page size then the <ServiceResponse> element shows the response code SUCCESS with the element <hasMoreRecords>true</hasMoreRecords> as shown below.

Follow the process below to obtain the first two the XML pages for an API request. Please apply the same logic to get all the next (n+1) pages until all records are returned. This is indicated when <hasMoreRecords>>false</hasMoreRecords>.

Request 1:

Search for tags that have a name containing the string “Windows”. The request in the POST data file “file.xml” defines this search criteria.

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/1.0/search/am/tag" <
file.xml
```

Note: “file.xml” contains the request POST data.

Request POST data for Request 1:

```
<ServiceRequest>
  <preferences>
    <limitResults>5</limitResults>
  </preferences>
  <filters>
    <Criteria field="name" operator="CONTAINS">Windows</Criteria>
  </filters>
</ServiceRequest>
```

Response:

The number of records is greater than the default pagination value so the <ServiceResponse> element identifies the last ID of the object in the current page output.

```
<ServiceResponse ...>
  <responseCode>SUCCESS</responseCode>
  <COUNT>5</COUNT>
  <hasMoreRecords>true</hasMoreRecords>
  <lastId>123</lastId>
  <data>
    <!--here you will find 5 tag records-->
  </data>
</ServiceResponse>
```

Request 2:

To get the next page of results, you need to edit your service request in “file.xml” that will be passed to API request as a POST payload. According to the <lastId> element returned in the first page, you want the next page of results to start with the object ID 124 or greater.

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"  
--data-binary @-  
"https://qualysapi.qualys.com/qps/rest/1.0/search/am/tag" <  
file.xml
```

Request POST Data for Request 2:

You'll notice the operator field value is set to 123, which is the value returned in <lastId> of the previous page output. The GREATER operator is a logical "greater than" (it does not mean greater than or equal to).

```
<ServiceRequest>  
  <preferences>  
    <limitResults>5</limitResults>  
  </preferences>  
  <filters>  
    <Criteria field="name"  
operator="CONTAINS">Windows</Criteria>  
    <Criteria field="id" operator="GREATER">123</Criteria>  
  </filters>  
</ServiceRequest>
```

Setting the Custom Page Size

The service request needs to contain the <preferences> section with the <limitResults> parameter. For the <limitResults> parameter you can enter a value from 1 to 1,000.

```
<ServiceRequest>  
  <filters>  
    <Criteria> ... </Criteria>  
  </filters>  
  <preferences>  
    <limitResults>200</limitResults>  
  </preferences>  
</ServiceRequest>
```

Base URL to the Qualys API Server

The Qualys API documentation and sample code within it use the API server URL for the Qualys US Platform 1: `qualysapi.qualys.com`.

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

Account Location	API Server URL
Qualys US Platform 1	<code>https://qualysapi.qualys.com</code>
Qualys US Platform 2	<code>https://qualysapi.qg2.apps.qualys.com</code>
Qualys EU Platform	<code>https://qualysapi.qualys.eu</code>

Authentication

The application must authenticate using Qualys account credentials (user name and password) as part of the HTTP request. The credentials are transmitted using the “Basic Authentication Scheme” over HTTPS.

For more information, see the “Basic Authentication Scheme” section of RFC #2617:

`http://www.faqs.org/rfcs/rfc2617.html`

The exact method of implementing authentication will vary according to which programming language is used.

The allowed methods, POST and/or GET, for each API request are documented with each API call in this user guide.

Example

Basic authentication - recommended option:

```
curl -u "USERNAME:PASSWORD"  
https://qualysapi.qualys.com/qps/rest/1.0/count/am/hostasset
```

where `qualysapi.qualys.com` is the base URL to the Qualys API server where your account is located.

Tags API

The Tags API provides a suite of API functions for managing tags. The supported Tag operations are: create, update, get, count, find and delete.

Tag operations:

[Create Tag](#)

[Update Tag](#)

[Get Tag](#)

[Count Tags](#)

[Search Tags - By Asset Count](#)

[Search Tags - By Criteria](#)

[Delete Tag](#)

Create Tag

Create a new tag and optionally assign a parent tag. Children tags are only listed. Children tags may be modified by assigning a new parent to a child tag.

URL: `https://qualysapi.qualys.com/qps/rest/1.0/create/am/tag`

Methods allowed: POST

Permissions

Managers with Full Scope. Users without Full Scope must have these permissions:
Access Permission “API Access”
Tag Permission “Create User Tag”
Tag Permission “Modify Dynamic Tag Rules” (dynamic tag only)

Example

This example creates a new host asset.

Request:

```
curl -u "USERNAME:PASSWORD" -H "Content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/1.0/create/am/tag" <
file.xml
```

Note: “file.xml” contains the request POST data.

Request POST data (file.xml):

```
<ServiceRequest>
<data>
<Tag>
  <scope>USER</scope>
  <name><![CDATA[create tag test]]></name>
  <description><![CDATA[sample tag test]]></description>
  <dynamicTagEngine>NAME_CONTAINS</dynamicTagEngine>
  <dynamicTagRule><![CDATA[oempa-loempa.*]]></dynamicTagRule>
  <icon>square</icon>
  <srcOperatingSystemName>windoze</srcOperatingSystemName>
  <reindex>>false</reindex>
  <display>
    <foregroundColor>-7197</foregroundColor>
    <backgroundcolor>-3407872</backgroundcolor>
```

```
</display>  
<parent>  
  <id>737931</id>  
</parent>  
</Tag>  
</data>  
</ServiceRequest>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/1.0/am/tag.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <Tag>
      <id>737932</id>
      <uuid>25d17e83-f270-487f-b871-dac48581f4f7</uuid>
      <name>create tag child test</name>
      <description>sample tag child test</description>
      <icon>round</icon>
      <dynamicTagRule><![CDATA[8.8.8.8]]></dynamicTagRule>
      <dynamicTagEngine>NETWORK_RANGE</dynamicTagEngine>
      <scope>USER</scope>
      <reindex>>true</reindex>
      <display>
        <foregroundColor>-7197</foregroundColor>
        <backgroundColor>-3407872</backgroundColor>
      </display>
      <parent>
        <id>737931</id>
        <uuid>95574648-545d-4e15-b182-d53fcea1718</uuid>
        <name>create tag test</name>
        <description>sample tag test</description>
        <icon>square</icon>
        <dynamicTagRule><![CDATA[oompa-
loomp*. *]]></dynamicTagRule>
        <dynamicTagEngine>NAME_CONTAINS</dynamicTagEngine>
        <scope>USER</scope>
        <srcOperatingSystemName>windoze</srcOperatingSystemName>
        <reindex>>false</reindex>
        <networks>
          <count>0</count>
        </networks>
      </parent>
      <display>
        <foregroundColor>-7197</foregroundColor>
        <backgroundColor>-3407872</backgroundColor>
      </display>
    </Tag>
  </data>
</ServiceResponse>
```



```

</display>
<createdBy>
  <id>20043120</id>
  <username>bsmith</username>
  <firstName><![CDATA[Bill]]></firstName>
  <lastName><![CDATA[Dmith]]></lastName>
</createdBy>
<created>2012-12-27T15:54:39Z</created>
<updatedBy>
  <id>20043120</id>
  <username>bsmith</username>
  <firstName><![CDATA[Bill]]></firstName>
  <lastName><![CDATA[Smith]]></lastName>
</updatedBy>
<updated>2012-12-27T15:54:39Z</updated>
</parent>
<children>
  <count>0</count>
</children>
<createdBy>
  <id>20043120</id>
  <username>bsmith</username>
  <firstName><![CDATA[Bill]]></firstName>
  <lastName><![CDATA[Smith]]></lastName>
</createdBy>
<created>2012-12-27T15:54:39Z</created>
<updatedBy>
  <id>20043120</id>
  <username>bsmith</username>
  <firstName><![CDATA[Bill]]></firstName>
  <lastName><![CDATA[Smith]]></lastName>
</updatedBy>
<updated>2012-12-27T15:54:39Z</updated>
</Tag>
</data>
</ServiceResponse>

```

Update Tag

Update a specific tag's members and optionally assign a parent tag. Same members used in a create request also apply to an update request.

If the tag's `<parent></parent>` is empty (e.g. no tag ID is specified) the tag's parent will be set to null/cleared. If the tag's `<parent></parent>` is missing the parent relationship is unchanged.

URL: `https://qualysapi.qualys.com/qps/rest/1.0/update/am/tag/<id>`

Methods allowed: POST

Permissions

Managers with Full Scope. Users without Full Scope must have these permissions:
Access Permission "API Access"
Tag Permission "Edit User Tag"
Tag Permission "Modify Dynamic Tag Rules" (if applicable)

Example

This example updates a host asset.

Request:

```
curl -u "USERNAME:PASSWORD" -H "Content-type: text/xml" -X "POST" --data-binary @-"https://qualysapi.qualys.com/qps/rest/1.0/update/am/tag/737932" < file.xml
```

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

Request POST data (file.xml):

```
<ServiceRequest>
<data>
  <Tag>
    <parent></parent>
  </Tag>
</data>
</ServiceRequest>
```

Response:

```
<?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/1.0/am/tag.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <Tag>
      <id>737932</id>
      <uuid>25d17e83-f270-487f-b871-dac48581f4f7</uuid>
      <name>create and updated tag parent test</name>
      <description>updatesample tag child test</description>
      <icon>round</icon>
      <dynamicTagRule><![CDATA[8.8.8.8,8.8.4.4]]></dynamicTagRule>
      <dynamicTagEngine>NETWORK_RANGE</dynamicTagEngine>
      <scope>USER</scope>
      <srcOperatingSystemName>Windoze</srcOperatingSystemName>
      <reindex>>true</reindex>
      <networks>
        <count>0</count>
      </networks>
      <display>
        <foregroundColor>-7196</foregroundColor>
        <backgroundColor>-3407871</backgroundColor>
      </display>
      <children>
        <count>0</count>
      </children>
      <createdBy>
        <id>20043120</id>
        <username>bsmith</username>
        <firstName><![CDATA[Bill]]></firstName>
        <lastName><![CDATA[Smith]]></lastName>
      </createdBy>
      <created>2012-12-27T15:54:39Z</created>
      <updatedBy>
        <id>20043120</id>
        <username>bsmith</username>
        <firstName><![CDATA[Bill]]></firstName>
        <lastName><![CDATA[Smith]]></lastName>
      </updatedBy>
```

```
        <updated>2012-12-27T15:54:40Z</updated>  
    </Tag>  
</data>  
</ServiceResponse>
```

Get Tag

Fetch a tag by ID.

URL: `https://qualysapi.qualys.com/qps/rest/1.0/get/am/tag/<id>`

Methods allowed: GET

Permissions

Managers with Full Scope. Users without Full Scope must have these permissions:
Access Permission“API Access”

Example

Fetch the host asset ID 776819 and list host asset details.

Request:

```
curl -n -u "USERNAME:PASSWORD"  
"https://qualysapi.qualys.com/qps/rest/1.0/get/am/tag/737931"
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>  
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-  
instance"  
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs  
d/1.0/am/tag.xsd">  
  <responseCode>SUCCESS</responseCode>  
  <count>1</count>  
  <data>  
    <Tag>  
      <id>737931</id>  
      <uuid>95574648-545d-4e15-b182-d53fceabl718</uuid>  
      <name>create tag test</name>  
      <description>sample tag test</description>  
      <icon>square</icon>  
      <dynamicTagRule><![CDATA[oompa-loompa.*]]></dynamicTagRule>  
      <dynamicTagEngine>NAME_CONTAINS</dynamicTagEngine>  
      <scope>USER</scope>  
      <srcOperatingSystemName>windoze</srcOperatingSystemName>  
      <reindex>>false</reindex>  
      <networks>  
        <count>0</count>
```

```
</networks>
<display>
  <foregroundColor>-7197</foregroundColor>
  <backgroundcolor>-3407872</backgroundcolor>
</display>
<children>
  <count>1</count>
  <list>
    <Tag>
      <id>737932</id>
      <uuid>25d17e83-f270-487f-b871-dac48581f4f7</uuid>
      <name>create and updated tag parent test</name>
      <description>updatesample tag child test</description>
      <icon>round</icon>

<dynamicTagRule><![CDATA[8.8.8.8,8.8.4.4]]></dynamicTagRule>
  <dynamicTagEngine>NETWORK_RANGE</dynamicTagEngine>
  <scope>USER</scope>

<srcOperatingSystemName>Windoze</srcOperatingSystemName>
  <reindex>true</reindex>
  <networks>
    <count>0</count>
  </networks>
<display>
  <foregroundColor>-7196</foregroundColor>
  <backgroundcolor>-3407871</backgroundcolor>
</display>
<createdBy>
  <id>20043120</id>
  <username>bsmith</username>
  <firstName><![CDATA[Bill]]></firstName>
  <lastName><![CDATA[Smith]]></lastName>
</createdBy>
<created>2012-12-27T15:54:39Z</created>
<updatedBy>
  <id>20043120</id>
  <username>bsmith</username>
  <firstName><![CDATA[Bill]]></firstName>
  <lastName><![CDATA[Smith]]></lastName>
</updatedBy>
```

```
        <updated>2012-12-27T15:54:41Z</updated>
      </Tag>
    </list>
  </children>
  <createdBy>
    <id>20043120</id>
    <username>bsmith</username>
    <firstName><![CDATA[Bill]]></firstName>
    <lastName><![CDATA[Smith]]></lastName>
  </createdBy>
  <created>2012-12-27T15:54:39Z</created>
  <updatedBy>
    <id>20043120</id>
    <username>bsmith</username>
    <firstName><![CDATA[Bill]]></firstName>
    <lastName><![CDATA[Smith]]></lastName>
  </updatedBy>
  <updated>2012-12-27T15:54:39Z</updated>
</Tag>
</data>
</ServiceResponse>
```

Count Tags

Count the number of tags available to the user.

URL: `https://qualysapi.qualys.com/qps/rest/1.0/count/am/tag`

Methods allowed: GET

Permissions

Managers with Full Scope. Users without Full Scope must have these permissions:
Access Permission“API Access”

Example

Request:

```
curl -u "USERNAME:PASSWORD"  
"https://qualysapi.qualys.com/qps/rest/1.0/count/am/tag"
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>  
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-  
instance"  
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs  
d/1.0/am/tag.xsd">  
  <responseCode>SUCCESS</responseCode>  
  <count>52</count>  
</ServiceResponse>
```


Search Tags - By Asset Count

Find Tags by assetCount criteria. Use the special Service Request Criteria field “assetCount” and numeric operators (e.g. EQUALS and NOT EQUALS, GREATER, LESSER) to find tags with the specified asset count. To find unused tags compare the assetCount EQUALS 0. To find all assigned tags use the assetCount NOT EQUALS to 0 operator. Warning: the CONTAINS and IN operators are not supported by this query.

Note: assetCount includes assets assigned the tag or any of the tag’s children. (You can compare this with the count that appears in the Qualys Asset Management application. Go to ASSET > Tags and then select a tag to see Tag Details. The “Assigned assets” value is the number of assets assigned the selected tag or any of the tag’s children.)

URL: `https://qualysapi.qualys.com/qps/rest/1.0/search/am/tag`

Methods allowed: POST

Permissions

Managers with Full Scope. Users without Full Scope must have these permissions:
Access Permission “API Access”

Example

Find all asset tags currently in use (assetCount NOT EQUALS to 0). Response shows 23 asset tags are in use.

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/1.0/search/am/tag" <
file.xml
```

Request POST data (file.xml):

```
<ServiceRequest>
  <preferences>
    <limitResults>100</limitResults>
  </preferences>
  <filters>
    <Criteria field="assetCount" operator="NOT
EQUALS">0</Criteria>
  </filters>
</ServiceRequest>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/1.0/am/tag.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>23</count>
  <hasMoreRecords>>false</hasMoreRecords>
  <data>
    <Tag>
      <id>437878</id>
      <uuid>ab361279-e03d-435b-934f-101dd4a9ab3f</uuid>
      <name>Asset Search Tags</name>
      <dynamicTagRule><![CDATA[ return false;]]></dynamicTagRule>
      <dynamicTagEngine>GROOVY</dynamicTagEngine>
      <scope>USER</scope>
      <reservedType>ASSET_SEARCH_ROOT</reservedType>
      <reindex>>true</reindex>
      <networks>
        <count>0</count>
      </networks>
      <display/>
      <children>
        <count>3</count>
        <list>
          <Tag>
            <id>437880</id>
            . . . . .
```

Search Tags - By Criteria

Find tags according to selected criteria.

URL: `https://qualysapi.qualys.com/qps/rest/1.0/search/am/tag`

Methods allowed: POST

Permissions

Managers with Full Scope. Users without Full Scope must have these permissions:
Access Permission“API Access”

Example

Find an tags by name.

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/1.0/search/am/tag" <
file.xml
```

Request POST data (file.xml):

```
<ServiceRequest>
  <preferences>
    <limitResults>100</limitResults>
  </preferences>
  <filters>
    <Criteria field="name" operator="CONTAINS">create
tag</Criteria>
  </filters>
</ServiceRequest>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/1.0/am/tag.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <hasMoreRecords>false</hasMoreRecords>
```

```
<data>
  <Tag>
    <id>737931</id>
    <uuid>95574648-545d-4e15-b182-d53fcea1718</uuid>
    <name>create tag test</name>
    <description>sample tag test</description>
    <icon>square</icon>
    <dynamicTagRule><![CDATA[oompa-loompa.*]]></dynamicTagRule>
    <dynamicTagEngine>NAME_CONTAINS</dynamicTagEngine>
    <scope>USER</scope>
    <srcOperatingSystemName>windoze</srcOperatingSystemName>
    <reindex>>false</reindex>
    <networks>
      <count>0</count>
    </networks>
    <display>
      <foregroundColor>-7197</foregroundColor>
      <backgroundColor>-3407872</backgroundColor>
    </display>
    <children>
      <count>1</count>
      <list>
        <Tag>
          <id>737932</id>
          <uuid>25d17e83-f270-487f-b871-dac48581f4f7</uuid>
          <name>create and updated tag parent test</name>
          <description>updatedsample tag child test</description>
          <icon>round</icon>
        </Tag>
      </list>
    </children>
  </Tag>
</dynamicTagRule><![CDATA[8.8.8.8,8.8.4.4]]></dynamicTagRule>
  <dynamicTagEngine>NETWORK_RANGE</dynamicTagEngine>
  <scope>USER</scope>
</srcOperatingSystemName>Windoze</srcOperatingSystemName>
  <reindex>>true</reindex>
  <networks>
    <count>0</count>
  </networks>
  <display>
    <foregroundColor>-7196</foregroundColor>
    <backgroundColor>-3407871</backgroundColor>
  </display>
```

```
</display>
<createdBy>
  <id>20043120</id>
  <username>bsmith</username>
  <firstName><![CDATA[Bill]]></firstName>
  <lastName><![CDATA[Smith]]></lastName>
</createdBy>
<created>2012-12-27T15:54:39Z</created>
<updatedBy>
  <id>20043120</id>
  <username>bsmith</username>
  <firstName><![CDATA[Bill]]></firstName>
  <lastName><![CDATA[Smith]]></lastName>
</updatedBy>
<updated>2012-12-27T15:54:41Z</updated>
</Tag>
</list>
</children>
<createdBy>
  <id>20043120</id>
  <username>bsmith</username>
  <firstName><![CDATA[Bill]]></firstName>
  <lastName><![CDATA[Smith]]></lastName>
</createdBy>
<created>2012-12-27T15:54:39Z</created>
<updatedBy>
  <id>20043120</id>
  <username>bsmith</username>
  <firstName><![CDATA[Bill]]></firstName>
  <lastName><![CDATA[Smith]]></lastName>
</updatedBy>
<updated>2012-12-27T15:54:39Z</updated>
</Tag>
</data>
</ServiceResponse>
```

Delete Tag

Delete a tag by ID.

URL: `https://qualysapi.qualys.com/qps/rest/1.0/delete/am/tag/<id>`

Methods allowed: POST

Permissions

Managers with Full Scope. Users without Full Scope must have these permissions:
Access Permission “API Access”
Tag Permission “Delete User Tag”

Examples

Delete the tag that has the ID 737932.

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/1.0/delete/am/tag/737932"
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/1.0/am/tag.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <Tag>
      <id>737932</id>
    </Tag>
  </data>
</ServiceResponse>
```

Host Asset API

The Host Asset API provides a suite of API functions for managing host assets. The supported Host Asset operations are create, update, get, find, count and delete. The HostAsset members identify operating system, NetBIOS, tags, open ports, NICs, installed software, banned tags, EC2 source information and current vulnerabilities (all instances).

Using the Host Asset API a banned tag may be defined for a host, so the tag will never be applied to the host (even if it satisfies the conditions in the tag rule). This feature is not available using the Qualys user interface.

Host Asset operations:

[Create Host Asset](#)

[Update Host Asset](#)

[Get Host Asset](#)

[Count Host Assets](#)

[Search Assigned Host Assets - By ID](#)

[Search Hosts Assets - By Criteria](#)

[Delete Host Asset](#)

Create Host Asset

Create a new host asset (HostAsset) and optionally add, remove, set tags, open ports, NICs, and installed software. Banned tags and EC2 source information is listed and may not be modified. Current vulnerabilities (HostInstanceVulns) are only listed and may not be modified.

URL: `https://qualysapi.qualys.com/qps/rest/1.0/create/am/hostasset`

Methods allowed: POST

Permissions

Managers with Full Scope. Users without Full Scope must have:
Access Permission “API Access”
Asset Management Permission “Create Asset”

Example

This example creates a new host asset.

Request:

```
curl -u "USERNAME:PASSWORD" -H "Content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/1.0/create/am/hostasset" <
file.xml
```

Note: “file.xml” contains the request POST data.

Request POST data (file.xml):

```
<ServiceRequest>
<data>
  <HostAsset>
    <name><![CDATA[HostAsset test with NIC, Software, Open
Ports]]></name>
    <trackingMethod>DNSNAME</trackingMethod>
    <netbiosName>megabios-1234-sfg</netbiosName>
    <netbiosNetworkId>112233</netbiosNetworkId>
    <operatingSystem>Windoze</operatingSystem>
    <tags>
      <set>
        <Tag>
          <id>737933</id>
```



```

        </Tag>
    </set>
</tags>
<installedSoftwares>
    <set>
        <HostSoftware>
            <name>IIS</name>
            <type>web server</type>
            <version>v1.2.3-beta</version>
        </HostSoftware>
        <HostSoftware>
            <name>Flash</name>
            <type>web plugin</type>
            <version>v10.9.8</version>
        </HostSoftware>
    </set>
</installedSoftwares>
<nicAddresses>
    <set>
        <NetworkInterfaceAddress>
            <inetAddress>
                <hostName>google-public-dns-
a.google.com</hostName>
            </inetAddress>
            <interfaceId>12341234</interfaceId>
            <interfaceName>LAN4</interfaceName>
            <type>ELASTIC</type>
        </NetworkInterfaceAddress>
        <NetworkInterfaceAddress>
            <inetAddress>
                <hostName>google-public-dns-
b.google.com</hostName>
            </inetAddress>
            <interfaceId>5678568</interfaceId>
            <interfaceName>LAN8</interfaceName>
            <type>ELASTIC</type>
        </NetworkInterfaceAddress>
    </set>
</nicAddresses>
<openPorts>
    <set>

```

```
        <HostOpenPort>
            <port>80</port>
            <protocol>TCP</protocol>
        </HostOpenPort>
        <HostOpenPort>
            <port>22</port>
            <protocol>UDP</protocol>
        </HostOpenPort>
    </set>
</openPorts>
</HostAsset>
</data>
</ServiceRequest>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/1.0/am/hostasset.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <HostAsset>
      <id>776864</id>
      <name><![CDATA[HostAsset test with NIC, Software, Open
Ports]]></name>
      <assetType>HOST</assetType>
      <tags>
        <count>1</count>
        <list>
          <Tag>
            <id>737933</id>
            <uuid>bc8f83aa-eb1f-4a1d-b9ae-167c94e11d72</uuid>
            <name>create tag and add to hostasset test</name>
            <description>sample tag test</description>
            <icon>square</icon>
            <dynamicTagRule><![CDATA[HostAsset
test]]></dynamicTagRule>
            <dynamicTagEngine>NAME_CONTAINS</dynamicTagEngine>
            <scope>USER</scope>

          </Tag>
        </list>
      </tags>
      <srcOperatingSystemName>windoze</srcOperatingSystemName>
      <reindex>true</reindex>
      <networks>
        <count>0</count>
      </networks>
      <display>
        <foregroundColor>-7197</foregroundColor>
        <backgroundcolor>-3407872</backgroundcolor>
      </display>
      <createdBy>
        <id>20043120</id>
        <username>bsmith</username>
        <firstName><![CDATA[Bill]]></firstName>
      </createdBy>
    </HostAsset>
  </data>
</ServiceResponse>
```

```
        <lastName><![CDATA[Smith]]></lastName>
    </createdBy>
    <created>2012-12-27T17:52:45Z</created>
    <updatedBy>
        <id>20043120</id>
        <username>bsmith</username>
        <firstName><![CDATA[Bill]]></firstName>
        <lastName><![CDATA[Smith]]></lastName>
    </updatedBy>
    <updated>2012-12-27T17:52:45Z</updated>
</Tag>
</list>
</tags>
<bannedTags>
    <count>0</count>
</bannedTags>
<sourceInformation>
    <count>0</count>
</sourceInformation>
<deleted>>false</deleted>
<nicAddresses>
    <count>2</count>
    <list>
        <NetworkInterfaceAddress>
            <id>703</id>
            <inetAddress>
                <hostName>google-public-dns-a.google.com</hostName>
                <ipAddress>8.8.8.8</ipAddress>
            </inetAddress>
            <interfaceId>12341234</interfaceId>
            <interfaceName>LAN4</interfaceName>
            <type>ELASTIC</type>
        </NetworkInterfaceAddress>
        <NetworkInterfaceAddress>
            <id>704</id>
            <inetAddress>
                <hostName>google-public-dns-b.google.com</hostName>
                <ipAddress>8.8.4.4</ipAddress>
            </inetAddress>
            <interfaceId>5678568</interfaceId>
            <interfaceName>LAN8</interfaceName>
```

```

        <type>ELASTIC</type>
    </NetworkInterfaceAddress>
</list>
</nicAddresses>
<trackingMethod>DNSNAME</trackingMethod>
<netbiosName>megabios-1234-sfg</netbiosName>
<netbiosNetworkId>112233</netbiosNetworkId>
<operatingSystem>Windoze</operatingSystem>
<openPorts>
    <count>2</count>
    <list>
        <HostOpenPort>
            <id>1149</id>
            <port>80</port>
            <protocol>TCP</protocol>
            <updatedAtDate>2012-12-27T17:52:46Z</updatedAtDate>
        </HostOpenPort>
        <HostOpenPort>
            <id>1150</id>
            <port>22</port>
            <protocol>UDP</protocol>
            <updatedAtDate>2012-12-27T17:52:46Z</updatedAtDate>
        </HostOpenPort>
    </list>
</openPorts>
<installedSoftwares>
    <count>2</count>
    <list>
        <HostSoftware>
            <id>1160</id>
            <name>IIS</name>
            <type>web server</type>
            <version>v1.2.3-beta</version>
        </HostSoftware>
        <HostSoftware>
            <id>1161</id>
            <name>Flash</name>
            <type>web plugin</type>
            <version>v10.9.8</version>
        </HostSoftware>
    </list>

```

```
    </installedSoftwares>  
    <vulns>  
      <count>0</count>  
    </vulns>  
  </HostAsset>  
</data>  
</ServiceResponse>
```

Update Host Asset

Update a specific host asset's members and optionally add, remove, set tags, open ports, NICs, and installed software. Same members used in a create request also apply to an update request. Banned tags and EC2 source information is listed and may not be modified. Current vulnerabilities (HostInstanceVulns) are listed and may not be modified.

URL: `https://qualysapi.qualys.com/qps/rest/1.0/update/am/hostasset/<id>`

Methods allowed: POST

Permissions

Managers with Full Scope. Users without Full Scope must have these permissions:
Access Permission "API Access"
Asset Management Permission "Update Asset"

Example

This example updates a host asset.

Request:

```
curl -u "USERNAME:PASSWORD" -H "Content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/1.0/update/am/hostasset/776864" < file.xml
```

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

Request POST data (file.xml):

```
<ServiceRequest>
<data>
  <HostAsset>
    <id>776864</id>
    <name><![CDATA[updated HostAsset test with NIC, Software,
Open Ports]]></name>
    <trackingMethod>INSTANCE_ID</trackingMethod>
    <netbiosName>updated megabios-1234-sfg</netbiosName>
    <netbiosNetworkId>9112233</netbiosNetworkId>
    <operatingSystem>updated Windoze</operatingSystem>
    <tags>
      <remove>
```

```
        <Tag>
            <id>737933</id>
        </Tag>
    </remove>
</tags>
<installedSoftwares>
    <remove>
        <HostSoftware>
            <id>1160</id>
        </HostSoftware>
    </remove>
</installedSoftwares>
<nicAddresses>
    <remove>
        <NetworkInterfaceAddress>
            <id>703</id>
        </NetworkInterfaceAddress>
    </remove>
</nicAddresses>
<openPorts>
    <remove>
        <HostOpenPort>
            <id>1149</id>
        </HostOpenPort>
    </remove>
</openPorts>
</HostAsset>
</data>
</ServiceRequest>
```

Response:

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



```

    <id>776864</id>
    <name><![CDATA[updated HostAsset test with NIC, Software,
Open Ports]]></name>
    <assetType>HOST</assetType>
    <tags>
      <count>0</count>
    </tags>
    <bannedTags>
      <count>0</count>
    </bannedTags>
    <sourceInformation>
      <count>0</count>
    </sourceInformation>
    <deleted>false</deleted>
    <nicAddresses>
      <count>1</count>
      <list>
        <NetworkInterfaceAddress>
          <id>704</id>
          <inetAddress>
            <hostName>google-public-dns-b.google.com</hostName>
            <ipAddress>8.8.4.4</ipAddress>
          </inetAddress>
          <interfaceId>5678568</interfaceId>
          <interfaceName>LAN8</interfaceName>
          <type>ELASTIC</type>
        </NetworkInterfaceAddress>
      </list>
    </nicAddresses>
    <trackingMethod>INSTANCE_ID</trackingMethod>
    <netbiosName>updated megabios-1234-sfg</netbiosName>
    <netbiosNetworkId>9112233</netbiosNetworkId>
    <operatingSystem>updated Windoze</operatingSystem>
    <openPorts>
      <count>1</count>
      <list>
        <HostOpenPort>
          <id>1150</id>
          <port>22</port>
          <protocol>UDP</protocol>
          <updatedAtDate>2012-12-27T17:52:46Z</updatedAtDate>
        </HostOpenPort>
      </list>
    </openPorts>
  </HostAsset>

```

```
        </HostOpenPort>
    </list>
</openPorts>
<installedSoftwares>
    <count>1</count>
    <list>
        <HostSoftware>
            <id>1161</id>
            <name>Flash</name>
            <type>web plugin</type>
            <version>v10.9.8</version>
        </HostSoftware>
    </list>
</installedSoftwares>
<vulns>
    <count>0</count>
</vulns>
</HostAsset>
</data>
</ServiceResponse>
```

Get Host Asset

Fetch a host asset by ID, and list tags, open ports, NICs, installed software, banned tags, EC2 source information, and current vulnerabilities (HostInstanceVulns).

URL: `https://qualysapi.qualys.com/qps/rest/1.0/get/am/hostasset/<id>`

Methods allowed: GET

Permissions

Managers with Full Scope. Users without Full Scope must have:
Access Permission “API Access”
Asset Management Permission “Read Asset”

Example

Fetch the host asset ID 776819 and list host asset details.

Request:

```
curl -n -u "USERNAME:PASSWORD"  
"https://qualysapi.qualys.com/qps/rest/1.0/get/am/hostasset/776819"  
"
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>  
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-  
instance"  
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs  
d/1.0/am/hostasset.xsd">  
  <responseCode>SUCCESS</responseCode>  
  <count>1</count>  
  <data>  
    <HostAsset>  
      <id>776819</id>  
      <name><![CDATA[EC2-2014-Enterprise]]></name>  
      <hostName><![CDATA[ec2-50-19-55-121.compute-  
1.amazonaws.com]]></hostName>  
      <assetType>HOST</assetType>  
      <tags>  
        <count>2</count>  
        <list>  
          <Tag>
```

```
<id>737534</id>
<uuid>c2eafd96-4d61-43c9-b340-c5c6dffaafdc0</uuid>
<name>Example Tag Trees</name>
<scope>USER</scope>
<reindex>true</reindex>
<networks>
  <count>0</count>
</networks>
<display/>
<createdBy>
  <id>20043120</id>
  <username>bsmith</username>
  <firstName><![CDATA[Bill]]></firstName>
  <lastName><![CDATA[Smith]]></lastName>
</createdBy>
<created>2012-10-24T21:24:27Z</created>
<updatedBy>
  <id>20043120</id>
  <username>bsmith</username>
  <firstName><![CDATA[Bill]]></firstName>
  <lastName><![CDATA[Smith]]></lastName>
</updatedBy>
<updated>2012-10-24T21:27:49Z</updated>
</Tag>
<Tag>
  <id>737536</id>
  <uuid>a6f62f6e-98f6-4556-bcc6-06d02642e7b2</uuid>
  <name>Bad Tree</name>
  <scope>USER</scope>
  <reindex>true</reindex>
  <networks>
    <count>0</count>
  </networks>
  <display/>
  <updated>2012-10-24T21:25:41Z</updated>
</Tag>
</list>
</tags>
<bannedTags>
  <count>0</count>
</bannedTags>
```

```
<sourceInformation>
  <count>10</count>
  <list>
    <SourceInformation>
      <type>EC2</type>
      <name>AVAILABILITY_ZONE</name>
      <value>us-east-1a</value>
    </SourceInformation>
    <SourceInformation>
      <type>EC2</type>
      <name>DNS_PRIVATE</name>
      <value>ip-10-127-162-46.ec2.internal</value>
    </SourceInformation>
    <SourceInformation>
      <type>EC2</type>
      <name>DNS_PUBLIC</name>
      <value>ec2-50-19-55-121.compute-1.amazonaws.com</value>
    </SourceInformation>
    <SourceInformation>
      <type>EC2</type>
      <name>INSTANCE_ID</name>
      <value>i-59bc3426</value>
    </SourceInformation>
    <SourceInformation>
      <type>EC2</type>
      <name>INSTANCE_TYPE</name>
      <value>t1.micro</value>
    </SourceInformation>
    <SourceInformation>
      <type>EC2</type>
      <name>IMAGE_ID</name>
      <value>ami-218f0848</value>
    </SourceInformation>
    <SourceInformation>
      <type>EC2</type>
      <name>IP_PRIVATE</name>
      <value>10.127.162.46</value>
    </SourceInformation>
    <SourceInformation>
      <type>EC2</type>
      <name>IP_PUBLIC</name>
```

```
        <value>50.19.55.121</value>
    </SourceInformation>
    <SourceInformation>
        <type>EC2</type>
        <name>MONITORING_ENABLED</name>
        <value>>false</value>
    </SourceInformation>
    <SourceInformation>
        <type>EC2</type>
        <name>STATE</name>
        <value>RUNNING</value>
    </SourceInformation>
</list>
</sourceInformation>
<deleted>>false</deleted>
<nicAddresses>
    <count>2</count>
    <list>
        <NetworkInterfaceAddress>
            <id>607</id>
            <hostName>ec2-50-19-55-121.compute-
1.amazonaws.com</hostName>
            <inetAddress>
                <hostName>ec2-50-19-55-121.compute-
1.amazonaws.com</hostName>
                <ipAddress>50.19.55.121</ipAddress>
            </inetAddress>
            <interfaceName>EC2 Public IP Address</interfaceName>
            <type>PUBLIC</type>
        </NetworkInterfaceAddress>
        <NetworkInterfaceAddress>
            <id>605</id>
            <hostName>ip-10-127-162-46.ec2.internal</hostName>
            <inetAddress>
                <hostName>10.127.162.46</hostName>
                <ipAddress>10.127.162.46</ipAddress>
            </inetAddress>
            <interfaceName>EC2 Private IP Address</interfaceName>
            <type>PRIVATE</type>
        </NetworkInterfaceAddress>
    </list>
```

```
</nicAddresses>
<trackingMethod>INSTANCE_ID</trackingMethod>
<operatingSystem>PROD AMI for Amazon VPC</operatingSystem>
<openPorts>
  <count>0</count>
</openPorts>
<installedSoftwares>
  <count>0</count>
</installedSoftwares>
<vulns>
  <count>0</count>
</vulns>
</HostAsset>
</data>
</ServiceResponse>
```

Count Host Assets

Count the number of host assets available to the user.

URL: `https://qualysapi.qualys.com/qps/rest/1.0/count/am/hostasset`

Methods allowed: GET

Permissions

Managers with Full Scope. Users without Full Scope must have these permissions:
Access Permission “API Access”
Asset Management Permission “Read Asset”

Example

Request:

```
curl -u "USERNAME:PASSWORD"  
"https://qualysapi.qualys.com/qps/rest/1.0/count/am/hostasset"
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>  
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-  
instance"  
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs  
d/1.0/am/hostasset.xsd">  
  <responseCode>SUCCESS</responseCode>  
  <count>3175</count>  
</ServiceResponse>
```


Search Assigned Host Assets - By ID

Find assets that have an assigned tag by tag ID. Use the special Service Request Criteria field "assignedTagId" and the EQUALS and NOT EQUALS operator to find assets with the assigned tag, or not with the assigned tag. Only EQUALS and NOT EQUALS operators are supported by this query..

URL: `https://qualysapi.qualys.com/qps/rest/1.0/search/am/hostasset`

Methods allowed: POST

Permissions

Managers with Full Scope. Users without Full Scope must have these permissions:
Access Permission "API Access"
Asset Management Permission "Read Asset"

Example

Find a limit of 10 host assets by assigned tag. 10 host assets were found and there's more.

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --data-binary @-"https://qualysapi.qualys.com/qps/rest/1.0/search/am/hostasset" <file.xml
```

Request POST data (file.xml):

```
<ServiceRequest>
  <preferences>
    <limitResults>10</limitResults>
  </preferences>
  <filters>
    <Criteria field="assignedTagId"
operator="EQUALS">737522</Criteria>
  </filters>
</ServiceRequest>
```

Response:

```
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
```

```
d/1.0/am/hostasset.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>10</count>
  <hasMoreRecords>true</hasMoreRecords>
  <lastId>776710</lastId>
  <data>
    <HostAsset>
      <id>776700</id>
      <name><![CDATA[qa-vsca-550-1]]></name>
      <hostName><![CDATA[ec2-184-73-174-42.compute-
1.amazonaws.com]]></hostName>
      <assetType>HOST</assetType>
      <tags>
        <count>8</count>
        <list>
          <Tag>
.....
```

Search Hosts Assets - By Criteria

Find assets according to specified criteria and paging preferences. The Service Request Criteria allows the operator to evaluate value fields, like name.

URL: `https://qualysapi.qualys.com/qps/rest/1.0/search/am/hostasset`

Methods allowed: POST

Permissions

Managers with Full Scope. Users without Full Scope must have these permissions:

Access Permission “API Access”

Asset Management Permission “Read Asset”

Example

Find an asset by name.

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/1.0/search/am/hostasset" <
file.xml
```

Request POST data (file.xml):

```
<ServiceRequest>
  <preferences>
    <limitResults>100</limitResults>
  </preferences>
  <filters>
    <Criteria field="name" operator="CONTAINS">HostAsset
test</Criteria>
  </filters>
</ServiceRequest>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/1.0/am/hostasset.xsd">
```

```
<responseCode>SUCCESS</responseCode>
<count>1</count>
<hasMoreRecords>>false</hasMoreRecords>
<data>
  <HostAsset>
    <id>776824</id>
    <name><![CDATA[HostAsset test with NIC, Software, Open
Ports]]></name>
    <assetType>HOST</assetType>
    <tags>
      <count>1</count>
      <list>
        <Tag>
          <id>737484</id>
          <uuid>3f2f900e-4183-4cc3-90ea-9f808b31b558</uuid>
          <name>TagOne</name>
          <dynamicTagRule><![CDATA[//!asset.tags.find{it.name ==
"TagTwo"}
return false;]]></dynamicTagRule>
          <dynamicTagEngine>GROOVY</dynamicTagEngine>
          <scope>USER</scope>
          <reindex>>true</reindex>
          <networks>
            <count>0</count>
          </networks>
          <display>
            <foregroundColor>-3871</foregroundColor>
            <backgroundColor>-1282048</backgroundColor>
          </display>
          <createdBy>
            <id>20043120</id>
            <username>bsmith</username>
            <firstName><![CDATA[Bill]]></firstName>
            <lastName><![CDATA[Smith]]></lastName>
          </createdBy>
          <created>2012-08-13T14:46:29Z</created>
          <updatedBy>
            <id>20043120</id>
            <username>bsmith</username>
            <firstName><![CDATA[Bill]]></firstName>
            <lastName><![CDATA[Smith]]></lastName>
          </updatedBy>
        </Tag>
      </list>
    </tags>
  </HostAsset>
</data>
```

```
        </updatedBy>
        <updated>2012-10-19T12:30:02Z</updated>
    </Tag>
</list>
</tags>
<bannedTags>
    <count>0</count>
</bannedTags>
<sourceInformation>
    <count>0</count>
</sourceInformation>
<deleted>false</deleted>
<nicAddresses>
    <count>2</count>
    <list>
        <NetworkInterfaceAddress>
            <id>615</id>
            <inetAddress>
                <hostName>google-public-dns-a.google.com</hostName>
                <ipAddress>8.8.8.8</ipAddress>
            </inetAddress>
            <interfaceId>12341234</interfaceId>
            <interfaceName>LAN4</interfaceName>
            <type>ELASTIC</type>
        </NetworkInterfaceAddress>
        <NetworkInterfaceAddress>
            <id>616</id>
            <inetAddress>
                <hostName>google-public-dns-b.google.com</hostName>
                <ipAddress>8.8.4.4</ipAddress>
            </inetAddress>
            <interfaceId>5678568</interfaceId>
            <interfaceName>LAN8</interfaceName>
            <type>ELASTIC</type>
        </NetworkInterfaceAddress>
    </list>
</nicAddresses>
<trackingMethod>DNSNAME</trackingMethod>
<netbiosName>megabios-1234-sfg</netbiosName>
<netbiosNetworkId>112233</netbiosNetworkId>
<operatingSystem>Windoze</operatingSystem>
```

```
<openPorts>
  <count>3</count>
  <list>
    <HostOpenPort>
      <id>1056</id>
      <port>80</port>
      <protocol>TCP</protocol>
      <updatedAt>2012-12-21T13:28:32Z</updatedAt>
    </HostOpenPort>
    <HostOpenPort>
      <id>1057</id>
      <port>22</port>
      <protocol>UDP</protocol>
      <updatedAt>2012-12-21T13:28:32Z</updatedAt>
    </HostOpenPort>
    <HostOpenPort>
      <id>1058</id>
      <port>80</port>
      <protocol>TCP</protocol>
      <updatedAt>2012-12-21T13:34:32Z</updatedAt>
    </HostOpenPort>
  </list>
</openPorts>
<installedSoftwares>
  <count>3</count>
  <list>
    <HostSoftware>
      <id>1066</id>
      <name>IIS</name>
      <type>web server</type>
      <version>v1.2.3-beta</version>
    </HostSoftware>
    <HostSoftware>
      <id>1067</id>
      <name>Flash</name>
      <type>web plugin</type>
      <version>v10.9.8</version>
    </HostSoftware>
    <HostSoftware>
      <id>1068</id>
      <name>IIS</name>
```

```
        <type>web server</type>
        <version>v1.2.3-beta</version>
    </HostSoftware>
</list>
</installedSoftwares>
<vulns>
    <count>0</count>
</vulns>
</HostAsset>
</data>
</ServiceResponse>
```

Delete Host Asset

Delete a host asset by ID.

URL: `https://qualysapi.qualys.com/qps/rest/1.0/delete/am/hostasset/<id>`

Methods allowed: POST

Permissions

Managers with Full Scope. Users without Full Scope must have these permissions:
Access Permission “API Access”
Asset Management Permission “Delete Asset”

Example

Delete the asset that has the ID 1234.

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --data-binary @"https://qualysapi.qualys.com/qps/rest/1.0/delete/am/hostasset/776865"
```

Response:

```
?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/1.0/am/hostasset.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <HostAsset>
      <id>776865</id>
    </HostAsset>
  </data>
</ServiceResponse>
```


Asset API

The Asset API is a subset of the Host Asset API. The supported Asset operations are update, get, find, count, and delete. The Asset members identify name, tags, banned tags, and EC2 source information.

Using the Host Asset API a banned tag may be defined for a host, so the tag will never be applied to the host (even if it satisfies the conditions in the tag rule). This feature is not available using the Qualys user interface.

Asset operations:

[Update Asset](#)

[Get Asset](#)

[Count Assets](#)

[Search Assets - By SOLR Keyword](#)

[Search Assigned Assets - By ID](#)

[Search Assets - By Criteria](#)

[Delete Asset](#)

Update Asset

Update a specific asset's members and optionally add, remove, set tags. The asset's banned tags and EC2 source information is listed but may not be modified. The asset subclass additional members will not be displayed, for example these HostAsset members will not be displayed: operatingSystem, netbios, OpenPorts, NICs, and InstalledSoftware, HostInstanceVulns (current vulnerabilities), and others.

URL: `https://qualysapi.qualys.com/qps/rest/1.0/update/am/asset/<id>`

Methods allowed: POST

Permissions

Managers with Full Scope. Users without Full Scope must have these permissions:
Access Permission “API Access”
Asset Management Permission “Update Asset”

Example

This example adds a tag to an asset.

Request:

```
curl -u "USERNAME:PASSWORD" -H "Content-type: text/xml" -X "POST" --data-binary @-"https://qualysapi.qualys.com/qps/rest/1.0/update/am/asset/776888" < file.xml
```

Note: “file.xml” contains the request POST data.

Request POST data (file.xml):

```
<ServiceRequest>
<data>
  <Asset>
    <name><![CDATA[updated Asset test]]></name>
    <id>776888</id>
    <tags>
      <add>
        <Tag>
          <id>737959</id>
        </Tag>
      </add>
    </tags>
```

```

    </Asset>
</data>
</ServiceRequest>

```

Response:

```

<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/1.0/am/asset.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <Asset>
      <id>776888</id>
      <name><![CDATA[updated Asset test]]></name>
      <assetType>HOST</assetType>
      <tags>
        <count>1</count>
        <list>
          <Tag>
            <id>737959</id>
            <uuid>bda3ee1d-7d81-4d46-a3d3-8061fdf42bb1</uuid>
            <name>create tag and add to asset test</name>
            <description>sample tag asset test</description>
            <icon>square</icon>
            <dynamicTagRule><![CDATA[Asset test]]></dynamicTagRule>
            <dynamicTagEngine>NAME_CONTAINS</dynamicTagEngine>
            <scope>USER</scope>
            <srcOperatingSystemName>ubuntu</srcOperatingSystemName>
            <reindex>true</reindex>
            <networks>
              <count>0</count>
            </networks>
            <display>
              <foregroundColor>-7197</foregroundColor>
              <backgroundColor>-3407872</backgroundColor>
            </display>
            <createdBy>
              <id>20043120</id>
              <username>bsmith</username>
            </createdBy>
          </Tag>
        </list>
      </tags>
    </Asset>
  </data>
</ServiceResponse>

```

```
        <firstName><![CDATA[Bill]]></firstName>
        <lastName><![CDATA[Smith]]></lastName>
    </createdBy>
    <created>2012-12-28T12:16:27Z</created>
    <updatedBy>
        <id>20043120</id>
        <username>bsmith</username>
        <firstName><![CDATA[Bill]]></firstName>
        <lastName><![CDATA[Smith]]></lastName>
    </updatedBy>
    <updated>2012-12-28T12:16:27Z</updated>
</Tag>
</list>
</tags>
<bannedTags>
    <count>0</count>
</bannedTags>
<sourceInformation>
    <count>0</count>
</sourceInformation>
<deleted>>false</deleted>
</Asset>
</data>
</ServiceResponse>
```

Get Asset

Fetch an asset by ID and list tags, banned tags, and EC2 source information.

URL: `https://qualysapi.qualys.com/qps/rest/1.0/get/am/asset/<id>`

Methods allowed: GET

Permissions

Managers with Full Scope. Users without Full Scope must have these permissions:
Access Permission “API Access”
Asset Management Permission “Read Asset”

Example

Fetch the asset ID 776888 and list asset details.

Request:

```
curl -n -u "USERNAME:PASSWORD"  
"https://qualysapi.qualys.com/qps/rest/1.0/get/am/asset/776888"
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>  
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-  
instance"  
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs  
d/1.0/am/asset.xsd">  
  <responseCode>SUCCESS</responseCode>  
  <count>1</count>  
  <data>  
    <Asset>  
      <id>776888</id>  
      <name><![CDATA[Asset test with Tag]]></name>  
      <assetType>HOST</assetType>  
      <tags>  
        <count>1</count>  
        <list>  
          <Tag>  
            <id>737959</id>  
            <uuid>bda3ee1d-7d81-4d46-a3d3-8061fdf42bb1</uuid>  
            <name>create tag and add to asset test</name>
```

```
<description>sample tag asset test</description>
<icon>square</icon>
<dynamicTagRule><![CDATA[Asset test]]></dynamicTagRule>
<dynamicTagEngine>NAME_CONTAINS</dynamicTagEngine>
<scope>USER</scope>
<srcOperatingSystemName>ubuntu</srcOperatingSystemName>
<reindex>true</reindex>
<networks>
  <count>0</count>
</networks>
<display>
  <foregroundColor>-7197</foregroundColor>
  <backgroundColor>-3407872</backgroundColor>
</display>
<createdBy>
  <id>20043120</id>
  <username>bsmith</username>
  <firstName><![CDATA[Bill]]></firstName>
  <lastName><![CDATA[Smith]]></lastName>
</createdBy>
<created>2012-12-28T12:16:27Z</created>
<updatedBy>
  <id>20043120</id>
  <username>bsmith</username>
  <firstName><![CDATA[Bill]]></firstName>
  <lastName><![CDATA[Smith]]></lastName>
</updatedBy>
<updated>2012-12-28T12:16:27Z</updated>
</Tag>
</list>
</tags>
<bannedTags>
  <count>0</count>
</bannedTags>
<sourceInformation>
  <count>0</count>
</sourceInformation>
<deleted>false</deleted>
</Asset>
</data>
</ServiceResponse>
```

Count Assets

Count the number of assets available to the user.

URL: `https://qualysapi.qualys.com/qps/rest/1.0/count/am/asset`

Methods allowed: GET

Permissions

Managers with Full Scope. Users without Full Scope must have these permissions:
Access Permission “API Access”
Asset Management Permission “Read Asset”

Example

Request:

```
curl -u "USERNAME:PASSWORD"  
"https://qualysapi.qualys.com/qps/rest/1.0/count/am/asset"
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>  
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-  
instance"  
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs  
d/1.0/am/asset.xsd">  
  <responseCode>SUCCESS</responseCode>  
  <count>3175</count>  
</ServiceResponse>
```

Search Assets - By SOLR Keyword

Find assets that are returned from a SOLR keyword search, where the SOLR query is executed like the Qualys Asset Search Portal. Use the special Service Request Criteria field "solrKeyword" and the EQUALS operator to find assets matching the solrClient.getIdsByQuery() call. Only the EQUALS operator is supported by this query.

URL: `https://qualysapi.qualys.com/qps/rest/1.0/search/am/asset`

Methods allowed: POST

Permissions

Managers with Full Scope. Users without Full Scope must have these permissions:
Access Permission "API Access"
Asset Management Permission "Read Asset"

Example

Find assets that match the splunk keyword. One asset is found.

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --data-binary @- "https://qualysapi.qualys.com/qps/rest/1.0/search/am/asset" < file.xml
```

Request POST data (file.xml):

```
<ServiceRequest>
  <preferences>
    <limitResults>10</limitResults>
  </preferences>
  <filters>
    <Criteria field="solrKeyword"
operator="EQUALS">splunk</Criteria>
  </filters>
</ServiceRequest>
```

Response:

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



```
d/1.0/am/asset.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <hasMoreRecords>>false</hasMoreRecords>
  <data>
    <Asset>
      <id>532567</id>
      <name><![CDATA[Splunk - internal]]></name>
      <assetType>WEBAPP</assetType>
      <tags>
        <count>0</count>
      </tags>
      <bannedTags>
        <count>0</count>
      </bannedTags>
      <sourceInformation>
        <count>0</count>
      </sourceInformation>
      <deleted>>false</deleted>
    </Asset>
  </data>
</ServiceResponse>
```

Search Assigned Assets - By ID

Find assets that have an assigned tag by tag ID. Use the special Service Request Criteria field “assignedTagId” and the EQUALS and NOT EQUALS operator to find assets with the assigned tag, or not with the assigned tag. Only EQUALS and NOT EQUALS operators are supported by this query..

URL: `https://qualysapi.qualys.com/qps/rest/1.0/search/am/asset`

Methods allowed: POST

Permissions

Managers with Full Scope. Users without Full Scope must have these permissions:
Access Permission “API Access”
Asset Management Permission “Read Asset”

Example

Find a limit of 10 assets by assigned tag. 10 assets were found and there’s more.

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --data-binary @- "https://qualysapi.qualys.com/qps/rest/1.0/search/am/asset" < file.xml
```

Request POST data (file.xml):

```
<ServiceRequest>
  <preferences>
    <limitResults>10</limitResults>
  </preferences>
  <filters>
    <Criteria field="assignedTagId"
operator="EQUALS">737522</Criteria>
  </filters>
</ServiceRequest>
```

Response:

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

```
d/1.0/am/asset.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>10</count>
  <hasMoreRecords>true</hasMoreRecords>
  <lastId>776710</lastId>
  <data>
    <Asset>
      <id>776700</id>
      <name><![CDATA[qa-vsca-550-1]]></name>

    <hostName><![CDATA[ec2-184-73-174-42.compute-1.amazonaws.com]]></h
ostName>
      <assetType>HOST</assetType>
      <tags>
        <count>8</count>
        <list>
          <Tag>
            <id>737524</id>
          </Tag>
        </list>
      </tags>
    </Asset>
  </data>
</response>
.....
```

Search Assets - By Criteria

Find assets according to specified criteria and paging preferences. The Service Request Criteria allows the operator to evaluate value fields, like name.

URL: `https://qualysapi.qualys.com/qps/rest/1.0/search/am/asset`

Methods allowed: POST

Permissions

Managers with Full Scope. Users without Full Scope must have these permissions:
Access Permission "API Access"
Asset Management Permission "Read Asset"

Example

Find an asset by name.

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/1.0/search/am/asset" <
file.xml
```

Request POST data (file.xml):

```
<ServiceRequest>
  <preferences>
    <limitResults>100</limitResults>
  </preferences>
  <filters>
    <Criteria field="name" operator="CONTAINS">updated
Asset</Criteria>
  </filters>
</ServiceRequest>
```

Response:

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

```
<count>1</count>
<hasMoreRecords>>false</hasMoreRecords>
<data>
  <Asset>
    <id>776888</id>
    <name><![CDATA[updated Asset test]]></name>
    <assetType>HOST</assetType>
    <tags>
      <count>1</count>
      <list>
        <Tag>
          <id>737959</id>
          <uuid>bda3eeld-7d81-4d46-a3d3-8061fdf42bb1</uuid>
          <name>create tag and add to asset test</name>
          <description>sample tag asset test</description>
          <icon>square</icon>
          <dynamicTagRule><![CDATA[Asset test]]></dynamicTagRule>
          <dynamicTagEngine>NAME_CONTAINS</dynamicTagEngine>
          <scope>USER</scope>
          <srcOperatingSystemName>ubuntu</srcOperatingSystemName>
          <reindex>>true</reindex>
          <networks>
            <count>0</count>
          </networks>
          <display>
            <foregroundColor>-7197</foregroundColor>
            <backgroundColor>-3407872</backgroundColor>
          </display>
          <createdBy>
            <id>20043120</id>
            <username>bsmith</username>
            <firstName><![CDATA[Bill]]></firstName>
            <lastName><![CDATA[Smith]]></lastName>
          </createdBy>
          <created>2012-12-28T12:16:27Z</created>
          <updatedBy>
            <id>20043120</id>
            <username>bsmith</username>
            <firstName><![CDATA[Bill]]></firstName>
            <lastName><![CDATA[Smith]]></lastName>
          </updatedBy>
        </Tag>
      </list>
    </tags>
  </Asset>
</data>
```

```
        <updated>2012-12-28T12:16:27Z</updated>
    </Tag>
</list>
</tags>
<bannedTags>
    <count>0</count>
</bannedTags>
<sourceInformation>
    <count>0</count>
</sourceInformation>
<deleted>false</deleted>
</Asset>
</data>
</ServiceResponse>
```

Delete Asset

Delete an asset by ID.

URL: `https://qualysapi.qualys.com/qps/rest/1.0/delete/am/asset/<id>`

Methods allowed: POST

Permissions

Managers with Full Scope. Users without Full Scope must have these permissions:
Access Permission “API Access”
Asset Management Permission “Delete Asset”

Examples

Delete the asset that has the ID 1234.

Request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --data-binary @-"https://qualysapi.qualys.com/qps/rest/1.0/delete/am/asset/1234"
```

Response:

```
<?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/1.0/am/asset.xsd">  
  <responseCode>SUCCESS</responseCode>  
  <count>1</count>  
  <data>  
    <Asset>  
      <id>776888</id>  
    </Asset>  
  </data>  
</ServiceResponse>
```

