



Qualys Cloud Platform v2.x

API Release Notes

Version 2.37

February 20, 2019

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

[Asset Management and Tagging API: Fetch AWS Account ID](#)

[Asset Management and Tagging API: Activate EC2 Assets in CertView Module](#)

[Web Application Scanning API: Send Email only on completion of Multi-Scan](#)

URL to the Qualys API Server

Qualys maintains multiple Qualys platforms. 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	https://qualysapi.qualys.com
Qualys US Platform 2	https://qualysapi.qg2.apps.qualys.com
Qualys US Platform 3	https://qualysapi.qg3.apps.qualys.com
Qualys EU Platform 1	https://qualysapi.qualys.eu
Qualys EU Platform 2	https://qualysapi.qg2.apps.qualys.eu
Qualys India Platform 1	https://qualysapi.qg1.apps.qualys.in
Qualys Private Cloud Platform	<a href="https://qualysapi.<customer_base_url>">https://qualysapi.<customer_base_url>

The Qualys API documentation and sample code use the API server URL for the Qualys US Platform 1. If your account is located on another platform, please replace this URL with the appropriate server URL for your account.

Asset Management and Tagging API: Fetch AWS Account ID

API affected	/qps/rest/2.0/get/am/assetdataconnector/<id> /qps/rest/2.0/search/am/assetdataconnector /qps/rest/2.0/get/am/awsassetdataconnector/<id> /qps/rest/2.0/search/am/awsassetdataconnector
New or Updated APIs	Updated
DTD or XSD changes	Yes

The Asset Management and Tagging API has been updated to fetch the AWS Account ID for Asset Data Connectors. You can fetch the AWS Account ID while getting the connector information and search for connectors using a particular AWS Account ID.

Input Parameters

New input parameter for searching with AWS Account ID is described below.

Parameter	Description
awsAccountId={value}	(long) Specify the AWS account ID an asset data connector is connecting to.

Sample 1 - Get connector information

API request:

```
curl -n -u "USERNAME:PASSWORD"  
"https://qualysapi.qualys.com/qps/rest/2.0/get/am/assetdataconnector/75602"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8"?>  
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xsi:noNamespaceSchemaLocation="http://qualysapi.qualys.com/qps/xsd/2.0/am  
/asset_data_connector.xsd">  
  <responseCode>SUCCESS</responseCode>  
  <count>1</count>  
  <data>  
    <AssetDataConnector>  
      <id>75602</id>  
      <name>connector1</name>  
      <awsAccountId>205767712438</awsAccountId>  
      <description>New connector</description>  
      <lastSync>2019-02-08T15:52:05Z</lastSync>  
      <connectorState>FINISHED_SUCCESS</connectorState>  
      <type>AWS</type>  
      <disabled>>false</disabled>
```

```
        <isGovCloudConfigured>>false</isGovCloudConfigured>
        <isChinaConfigured>>false</isChinaConfigured>
    </AssetDataConnector>
</data>
</ServiceResponse>
```

Sample 2 - Search connectors with an AWS Account ID

API request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --data-
binary @-
"https://qualysapi.qualys.com/qps/rest/2.0/search/am/assetdataconnector"
< file.xml
```

Note: file.xml contains request POST data

Request POST data

```
<?xml version="1.0" encoding="UTF-8" ?>
<ServiceRequest>
<filters>
<Criteria field="awsAccountId" operator="EQUALS">205767712438</Criteria>
</filters>
</ServiceRequest>
```

XML output:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://qualysapi.qualys.com/qps/xsd/2.0/am
/asset_data_connector.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <hasMoreRecords>>false</hasMoreRecords>
    <data>
        <AssetDataConnector>
            <id>75602</id>
            <name>connector1</name>
            <awsAccountId>205767712438</awsAccountId>
            <description>New connector</description>
            <lastSync>2019-02-08T15:52:05Z</lastSync>
            <connectorState>FINISHED_SUCCESS</connectorState>
            <type>AWS</type>
            <disabled>>false</disabled>
            <isGovCloudConfigured>>false</isGovCloudConfigured>
            <isChinaConfigured>>false</isChinaConfigured>
        </AssetDataConnector>
    </data>
</ServiceResponse>
```

XSD update

New element `awsAccountId` added to **asset_data_connector.xsd** and **awsassetdataconnector.xsd**.

Sample from asset_data_connector.xsd

```
...
<complexType name="AssetDataConnector">
  <sequence>
    <element name="id" type="long" minOccurs="0" maxOccurs="1" />
    <element name="name" minOccurs="0" maxOccurs="1" type="string" />
    <element name="awsAccountId" minOccurs="0" maxOccurs="1" type="string" />
    <element name="description" minOccurs="0" maxOccurs="1" type="string" />
    <element name="lastSync" minOccurs="0" maxOccurs="1" type="dateTime" />
    <element name="lastError" minOccurs="0" maxOccurs="1" type="string" />
  
```

Asset Management and Tagging API: Activate EC2 Assets in CertView Module

API affected	<code>/qps/rest/2.0/get/am/assetdataconnector/<id></code> <code>/qps/rest/2.0/update/am/assetdataconnector</code> <code>/qps/rest/2.0/get/am/awsassetdataconnector/<id></code> <code>/qps/rest/2.0/update/am/awsassetdataconnector</code> <code>/qps/rest/2.0/create/am/awsassetdataconnector</code>
New or Updated APIs	Updated
XSD changes	Yes

The Asset Management and Tagging API has been updated to add a new connector for the CertView module for AWS Asset Data Connectors.

You create a connector for CertView or can fetch the CertView connector details once it is enabled in the ActivationModule tag.

Sample 1 - Update connector to include CertView module

API request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --data-binary @-
"https://qualysapi.qualys.com/qps/rest/2.0/update/am/awsassetdataconnector/80201"
< file.xml
```

Note: file.xml contains request POST data

Request POST data:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ServiceRequest>
  <data>
    <AwsAssetDataConnector>
      <name>new connector-edit</name>
      <activation>
        <add>
          <ActivationModule>CERTVIEW</ActivationModule>
        </add>
      </activation>
    </AwsAssetDataConnector>
  </data>
</ServiceRequest>
```

XML output:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://v-qps1.dev.qualys.com:8080/portal-
api/xsd/2.0/am/aws_asset_data_connector.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <AwsAssetDataConnector>
      <id>80201</id>
    </AwsAssetDataConnector>
  </data>
</ServiceResponse>
```

Sample 2 - Create connectors to include CERTVIEW module

API request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --data-
binary @-
"https://qualysapi.qualys.com/qps/rest/2.0/create/am/awsassetdataconnecto
r/" < file.xml
```

Note: file.xml contains request POST data

Request POST data

```
<ServiceRequest>
  <data>
    <AwsAssetDataConnector>
      <name>ARN_Global</name>
      <arn>arn:aws:iam::705355653965:role/Demo-PODs</arn>
      <externalId></externalId>
      <allRegions>true</allRegions>
      <activation>
        <set>
          <ActivationModule>VM</ActivationModule>
          <ActivationModule>CERTVIEW</ActivationModule>
        </set>
      </activation>
    </AwsAssetDataConnector>
  </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/2.0/a
```

```
m/aws_asset_data_connector.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <AwsAssetDataConnector>
      <id>566601</id>
      <name>ARN_Global</name>
      <awsAccountId>705355653965</awsAccountId>
      <connectorState>QUEUED</connectorState>
      <type>AWS</type>
      <activation>
        <list>
          <ActivationModule>CERTVIEW</ActivationModule>
          <ActivationModule>VM</ActivationModule>
        </list>
      </activation>
      <disabled>>false</disabled>
      <isGovCloudConfigured>>false</isGovCloudConfigured>
      <isChinaConfigured>>false</isChinaConfigured>
      <arn>arn:aws:iam::705355653965:role/Demo-PODs</arn>
      <externalId>1550261312725</externalId>
      <qualysAwsAccountId>383031258652</qualysAwsAccountId>
      <authRecord/>
      <endpoints>
        <list>
          <AwsEndpointSimple>
            <regionCode>us-west-1</regionCode>
          </AwsEndpointSimple>
          <AwsEndpointSimple>
            <regionCode>ap-northeast-1</regionCode>
          </AwsEndpointSimple>
          <AwsEndpointSimple>
            <regionCode>eu-west-1</regionCode>
          </AwsEndpointSimple>
          <AwsEndpointSimple>
            <regionCode>eu-central-1</regionCode>
          </AwsEndpointSimple>
          <AwsEndpointSimple>
            <regionCode>ap-southeast-1</regionCode>
          </AwsEndpointSimple>
          <AwsEndpointSimple>
            <regionCode>us-east-1</regionCode>
          </AwsEndpointSimple>
          <AwsEndpointSimple>
            <regionCode>ca-central-1</regionCode>
          </AwsEndpointSimple>
          <AwsEndpointSimple>
            <regionCode>eu-west-2</regionCode>
          </AwsEndpointSimple>
        </list>
      </endpoints>
    </AwsAssetDataConnector>
  </data>
</response>
```



```
<AwsEndpointSimple>
  <regionCode>ap-southeast-2</regionCode>
</AwsEndpointSimple>
<AwsEndpointSimple>
  <regionCode>sa-east-1</regionCode>
</AwsEndpointSimple>
<AwsEndpointSimple>
  <regionCode>ap-northeast-2</regionCode>
</AwsEndpointSimple>
<AwsEndpointSimple>
  <regionCode>ap-south-1</regionCode>
</AwsEndpointSimple>
<AwsEndpointSimple>
  <regionCode>us-west-2</regionCode>
</AwsEndpointSimple>
<AwsEndpointSimple>
  <regionCode>us-east-2</regionCode>
</AwsEndpointSimple>
<AwsEndpointSimple>
  <regionCode>eu-west-3</regionCode>
</AwsEndpointSimple>
</list>
</endpoints>
<allRegions>true</allRegions>
</AwsAssetDataConnector>
</data>
</ServiceResponse>
```

Sample 3 - Get CertView connector information

API request:

```
curl -n -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/qps/rest/2.0/get/am/assetdataconnector/78801"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://v-qps1.dev.qualys.com:8080/portal-api/xsd/2.0/am/asset_data_connector.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <AssetDataConnector>
      <id>78801</id>
      <name>cv</name>
      <awsAccountId>383031258652</awsAccountId>
```

```
<lastSync>2019-02-12T23:58:05Z</lastSync>
<connectorState>FINISHED_SUCCESS</connectorState>
<type>AWS</type>
<activation>
  <list>
    <ActivationModule>CERTVIEW</ActivationModule>
    <ActivationModule>VM</ActivationModule>
  </list>
</activation>
<disabled>>false</disabled>
<isGovCloudConfigured>>false</isGovCloudConfigured>
<isChinaConfigured>>false</isChinaConfigured>
</AssetDataConnector>
</data>
</ServiceResponse>
```

XSD update (asset_data_connector.xsd)

New value CERTVIEW is added to ActivationModule tag.

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://am.oxm.api.portal.qualys.com/v2"
  xmlns:tns="http://am.oxm.api.portal.qualys.com/v2"
  elementFormDefault="qualified">

  <include schemaLocation="tag.xsd" />
  <include schemaLocation="asset_source.xsd" />

  <simpleType name="AssetDataConnectorType">
    <restriction base="string">
      <enumeration value="AWS" />
    </restriction>
  </simpleType>

  ...
  <simpleType name="ActivationModule">
    <restriction base="string">
      <enumeration value="VM" />
      <enumeration value="PC" />
      <enumeration value="SCA" />
      <enumeration value="CERTVIEW" />
    </restriction>
  </simpleType>
  ...
```

Web Application Scanning API: Send Email only on completion of Multi-Scan

API affected	/qps/rest/3.0/get/was/wasscan/<id> /qps/rest/3.0/launch/was/wasscan /qps/rest/3.0/get/was/wasscanschedule/<id> /qps/rest/3.0/create/was/wasscanschedule/ /qps/rest/3.0/update/was/wasscanschedule/<id>
New or Updated APIs	Updated
DTD or XSD changes	Yes

We have now added a new parameter for a multi-scan to configure when the email should be sent: completion of multi-scan or completion of individual scan in a multi-scan.

The new option is available only when:

- Scan involves multiple web applications (multi-scan)
- Email notification is enabled for the multi-scan.

Input Parameters

New input parameter to configure email notification to be sent at the completion of entire multi-scan or completion of individual scan in a multi-scan.

Parameter	Description
sendOneMail (Boolean)	Set to true to send one email upon multi-scan completion. Set to false to send one email upon completion of each individual scan. Example: <sendOneMail>true</sendOneMail> Note: sendOneMail is valid only when sendMail = true for a multi-scan (multiple web applications being scanned). If sendMail is set to false, sendOneMail will be ignored.

Sample 1 - Launch a multi-scan

Let us launch a scan that allows to send one email on completion of multi-scan (not for each individual scan in the group).

API request:

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

Request POST Data (file.xml):

```
<ServiceRequest>
  <data>
    <WasScan>
      <name><![CDATA[New Scan]]></name>
      <type>VULNERABILITY</type>
      <target>
        <webApps>
          <set>
            <WebApp><id>8389207</id></WebApp>
            <WebApp><id>8389244</id></WebApp>
          </set>
        </webApps>
        <scannerAppliance>
          <type>EXTERNAL</type>
        </scannerAppliance>
      </target>
      <profile>
        <id>2337683</id>
      </profile>
      <sendOneMail>true</sendOneMail>
    </WasScan>
  </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/wasscan.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WasScan>
      <id>3456140</id>
    </WasScan>
  </data>
</ServiceResponse>
```

Sample 2 - Get details of a multi-scan

Let us get scan details of a multi-scan.

API request:

```
curl -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/qps/rest/3.0/get/was/wasscan/3456140"
```

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/wasscan.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WasScan>
      <id>3456140</id>
      <name>
        <![CDATA[New Scan]]>
      </name>
      <reference>was/1550052847033.4677658</reference>
      <type>VULNERABILITY</type>
      <mode>API</mode>
      <multi>true</multi>
    ...
      <launchedDate>2019-02-13T10:14:07Z</launchedDate>
      <launchedBy>
        <id>68875289</id>
        <username>quays_js</username>
        <firstName>
          <![CDATA[John]]>
        </firstName>
        <lastName>
          <![CDATA[Smith]]>
        </lastName>
      </launchedBy>
      <status>FINISHED</status>
      <scanDuration>10710</scanDuration>
      <sendMail>true</sendMail>
      <sendOneMail>true</sendOneMail>
      <enableWAFAuth>false</enableWAFAuth>
    </WasScan>
  </data>
</ServiceResponse>
```

Sample 3 - Create a scan schedule

Let us create a schedule for a multi-scan to configure mail to be sent at the completion of scan (and not after completion of individual scans).

API request:

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

file.xml

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

Request POST Data (file.xml):

```
<ServiceRequest>
<data>
  <WasScanSchedule>
    <name><![CDATA[New MultiSchedule]]></name>
    <type>VULNERABILITY</type>
    <active>true</active>
    <scheduling>
      <startDate>2019-02-13T12:40:27Z</startDate>
      <timeZone>
        <code>Asia/Kolkata</code>
        <offset>+05:30</offset>
      </timeZone>
      <occurrenceType>ONCE</occurrenceType>
    </scheduling>
    <notification>
      <active>true</active>
      <delay>
        <nb>1</nb>
        <scale>DAY</scale>
      </delay>
    <message><![CDATA[Qualys scan is scheduled to start soon]]></message>
    </notification>
    <target>
      <webApps>
        <set>
          <WebApp><id>8389207</id></WebApp>
          <WebApp><id>8389244</id></WebApp>
        </set>
      </webApps>
      <scannerAppliance>
        <type>EXTERNAL</type>
      </scannerAppliance>
      <cancelOption>SPECIFIC</cancelOption>
      <authRecordOption>DEFAULT</authRecordOption>
      <profileOption>SPECIFIC</profileOption>
      <scannerOption>DEFAULT</scannerOption>
      <randomizeScan>true</randomizeScan>
      <useDnsOverride>true</useDnsOverride>
    </target>
    <profile><id>2337683</id></profile>
    <sendOneMail>true</sendOneMail>
  </WasScanSchedule>
</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/wasscanschedule.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WasScanSchedule>
      <id>865156</id>
      <name>
        <![CDATA[New MultiSchedule]]>
      </name>
      <owner>
        <id>68875289</id>
        <username>user_john</username>
        <firstName>
          <![CDATA[John]]>
        </firstName>
        <lastName>
          <![CDATA[Doe]]>
        </lastName>
      </owner>
      <active>false</active>
      <multi>true</multi>
      <type>VULNERABILITY</type>
      <target>
        <webApps>
          <list>
            ...
          </list>
        </webApps>
      </target>
      <sendMail>true</sendMail>
      <sendOneMail>true</sendOneMail>
      <enableWAFAuth>false</enableWAFAuth>
    </WasScanSchedule>
  </data>
</ServiceResponse>
```

Sample 4 - Update a schedule

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/wasscanschedule/865
156" < file.xml
Note: "file.xml" contains the request POST data.
```

Request POST Data (file.xml):

```
<ServiceRequest>
<data>
  <WasScanSchedule>
    <sendOneMail>false</sendOneMail>
  </WasScanSchedule>
</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/wasscanschedule.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WasScanSchedule>
      <id>865156</id>
    </WasScanSchedule>
  </data>
</ServiceResponse>
```

Sample 5 - Get schedule details

API request:

```
curl -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/qps/rest/3.0/get/was/wasscanschedule
/865156"
```

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/wasscanschedule.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WasScanSchedule>
      <id>856156</id>
      <name><![CDATA[John Doe]]></name>
      <owner>
        <id>68875289</id>
        <username>user_john</username>
        <firstName>
```



```

        <![CDATA[John]]>
      </firstName>
      <lastName>
        <![CDATA[Doe]]>
      </lastName>
    </owner>
    <active>false</active>
    <multi>true</multi>
    <type>VULNERABILITY</type>
    <target>
      <webApps>
        <list>
          <WebApp>
...
        </updatedBy>
        <sendMail>true</sendMail>
        <sendOneMail>true</sendOneMail>
        <enableWAFAuth>false</enableWAFAuth>
      </WasScanSchedule>
    </data>
  </ServiceResponse>

```

XSD Update

Changes in wasscan.xsd (/qps/xsd/3.0/was/wasscan.xsd) and wasscanschedule.xsd (/qps/xsd/3.0/was/wasscanschedule.xsd).

Scan (/qps/xsd/3.0/was/wasscan.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:element name="igs" type="WasScanIgList" minOccurs="0"/>

```

```
    <xs:element name="sendMail" type="xs:boolean" default="true"
minOccurs="0"/>
    <xs:element name="sendOneMail" type="xs:boolean" default="false"
minOccurs="0"/>
    <xs:element name="enableWAFAuth" type="xs:boolean" default="false"
minOccurs="0"/>
    </xs:all>
  </xs:complexType>
  ...
    <xs:enumeration value="DEFAULT"/>
    <xs:enumeration value="SPECIFIC"/>
  </xs:restriction>
</xs:simpleType>
</xs:schema>
```

Schedule (/qps/xsd/3.0/was/wasscanschedule.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="updatedAt" type="xs:dateTime" minOccurs="0"/>
        <xs:element name="updatedAtBy" type="User" minOccurs="0"/>
        <xs:element name="sendMail" type="xs:boolean" default="true"
minOccurs="0"/>
        <xs:element name="sendOneMail" type="xs:boolean" default="false"
minOccurs="0"/>
        <xs:element name="enableWAFAuth" type="xs:boolean" default="false"
minOccurs="0"/>
      </xs:all>
    </xs:complexType>
    ...
      <xs:enumeration value="DEFAULT"/>
      <xs:enumeration value="SPECIFIC"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```