

Xconf Data Service - REST APIs

- **Firmware Config**
 - Retrieve a list of firmware configs
 - Retrieve a single firmware config by id
 - Retrieve firmware configs by modelId
 - Create/update a firmware config
 - Delete a firmware config by id
- **IP rules**
 - Retrieve an ip rule list
 - Retrieve an ip rule by name
 - Create/update an ip rule
 - Delete an ip rule
- **Location filter**
 - Retrieve a location filter list
 - Retrieve a location filter by name
 - Create/update location filter
 - Delete location filter by name
 - Retrieve download location filter
 - Update download location filter
- **Environment model rules**
 - Retrieve an environment model rule list
 - Retrieve an environment model rule by name
 - Create/update an environment model rule
 - Delete an environment model rule
- **IP Filter**
 - Retrieve an IP filter list
 - Retrieve an ip filter by name
 - Create/update an IP filter
 - Delete IP filter
- **Percent filter**
 - Retrieve percent filter
 - Retrieve percent filter field values
 - Update percent filter
 - Retrieve EnvModelPercentages
 - Retrieve EnvModelPercentage by id
 - Create envModelPercentage
 - Update EnvModelPercentage
 - Delete envModelPercentage
- **Time filter**
 - Retrieve time filter list
 - Retrieve time filter by name
 - Create/update time filter
 - Delete time filter by name
- **Environment**
 - Retrieve an list of environments
 - Retrieve environment by id
 - Create an environment
 - Delete environment by id
- **IP address group**
 - Retrieve an IP address group list
 - Retrieve an IP address group by name
 - Retrieve an IP address group by IP
 - Create an IP address group
 - Add data to IP Address Group(dev in progress)
 - Delete data from IP Address Group(dev in progress)
 - Delete an IP address group by id
- **Mac rule**
 - Retrieve a mac rule list (legacy)
 - Retrieve a mac rule list (v2)
 - Retrieve mac rule by name (legacy)
 - Retrieve mac rule by name (v2)
 - Retrieve mac rule by mac address (legacy)
 - Retrieve mac rule by mac address (v2)
 - Create/update mac rule
 - Delete mac rule by name
- **Model**
 - Retrieve a model list
 - Retrieve model by id
 - Create model
 - Update Model Description
 - Delete model by id
- **NamespacedList**
 - Retrieve all NS lists
 - Retrieve an NS list by id:
 - Retrieve NS list by mac part
 - Create a NS list

- Add data to NS list
 - Delete data from NS list
 - Delete an NS list by id
- RebootImmediately filter
 - Retrieve an RI filter list
 - Retrieve and RI filter by rule name
 - Create/update an RI filter
 - Delete RI filter by name
- FirmwareRuleTemplate
 - Retrieve filtered templates
 - Import firmware rule templates
- FirmwareRule
 - Retrieve all firmware rules
 - Retrieve filtered firmware rules
 - Import firmware rule
- Feature
 - Retrieve all features
 - Retrieve filtered features
 - Import feature
 - Create feature
 - Update feature
 - Delete feature
- Feature Rule
 - Retrieve all feature rules
 - Retrieve filtered feature rules
 - Import feature rule
 - Create feature rule
 - Update feature rule
 - Delete feature rule
- Activation Minimum Version
 - Retrieve all activation minimum versions
 - Retrieve filtered activation minimum versions
 - Import activation version:
 - Create activation minimum version
 - Update activation minimum version
 - Delete activation minimum version



Note

A reference setup of the xconf dataservice is available at <https://xconf.rdkcentral.com:9092/> for the community members to get acquainted with xconf. You can replace the <host> , <port> and protocol as xconf.rdkcentral.com, 9092 and https respectively in the placeholders mentioned below in the request samples. Or if you have deployed your own server instance then replace the host and port accordingly.

Firmware Config

Retrieve a list of firmware configs

Request

```
GET http://<host>:<port>/queries/firmwares?applicationType={type}
```

Headers:

```
Accept = application/json OR application/xml
```

```
applicationType parameter is not required, default value is stb
```

```
Response: 200 OK OR 400 BAD REQUEST
```

Request: <http://<host>:<port>/queries/firmwares.json> OR <http://<host>:<port>/queries/firmwares>

JSON response:

```
{
  "id": "firmwareConfigId",
  "description": "FirmwareDescription",
  "supportedModelIds": [
    "MODELA"
  ],
  "firmwareFilename": "FirmwareFilename",
  "firmwareVersion": "FirmwareVersion"
}
```

XML response

```
<ArrayList>
  <item>
    <id>b65962b5-1481-4eed-a010-2abfa8c3bbfd</id>
    <description>_</description>
    <supportedModelIds>
      <supportedModelIds>YETST</supportedModelIds>
    </supportedModelIds>
    <firmwareFilename>_</firmwareFilename>
    <firmwareVersion>_</firmwareVersion>
  </item>
  <item>
    <id>f0b7b35b-4b8e-4a15-9d66-91c4b3d575d1</id>
    <description>prav_Firm</description>
    <supportedModelIds>
      <supportedModelIds>PX013ANM</supportedModelIds>
      <supportedModelIds>PX013ANC</supportedModelIds>
    </supportedModelIds>
    <firmwareFilename>PX013AN_2.1s11_VBN_HYBse-signed.bin</firmwareFilename>
    <firmwareVersion>PX013AN_2.1s11_VBN_HYBse-signed</firmwareVersion>
  </item>
</ArrayList>
```

Retrieve a single firmware config by id

GET <http://<host>:<port>/queries/firmwares/<firmwareConfigId>>

Headers:

Accept = application/json OR application/xml

Response: 200 OK

Request:

<http://<host>:<port>/queries/firmwares/b65962b5-1481-4eed-a010-2abfa8c3bbfd>

OR

<http://<host>:<port>/queries/firmwares/b65962b5-1481-4eed-a010-2abfa8c3bbfd.json>

JSON response

```
{
  "id": "b65962b5-1481-4eed-a010-2abfa8c3bbfd",
  "updated": 1440492963476,
  "description": "_-",
  "supportedModelIds": [
    "YETST"
  ],
  "firmwareDownloadProtocol": "tftp",
  "firmwareFilename": "_-",
  "firmwareVersion": "_-",
  "rebootImmediately": false
}
```

XML response

```
<FirmwareConfig>
  <id>b65962b5-1481-4eed-a010-2abfa8c3bbfd</id>
  <updated>1440492963476</updated>
  <description>_-</description>
  <supportedModelIds>
    <supportedModelIds>YETST</supportedModelIds>
  </supportedModelIds>
  <firmwareDownloadProtocol>tftp</firmwareDownloadProtocol>
  <firmwareFilename>_-</firmwareFilename>
  <firmwareVersion>_-</firmwareVersion>
  <rebootImmediately>>false</rebootImmediately>
</FirmwareConfig>
```

Retrieve firmware configs by modelId

GET <http://<host>:<port>/queries/firmwares/model/{modelId}?applicationType={type}>

Headers:

Accept = application/json OR application/xml

applicationType parameter is not required, default value is stb

Response: 200 OK OR 400 BAD REQUEST (if application type has a wrong value)

Request:

<http://<host>:<port>/queries/firmwares/model/YETST.json>

OR

<http://<host>:<port>/queries/firmwares/model/YETST>

JSON response

```
[{
  "id": "b65962b5-1481-4eed-a010-2abfa8c3bbfd",
  "updated": 1440492963476,
  "description": "_-",
  "supportedModelIds": [
    "YETST"
  ],
  "firmwareDownloadProtocol": "tftp",
  "firmwareFilename": "_-",
  "firmwareVersion": "_-",
  "rebootImmediately": false
}]
```

XML response

```
<ArrayList>
  <item>
    <id>b65962b5-1481-4eed-a010-2abfa8c3bbfd</id>
    <description>_</description>
    <supportedModelIds>
      <supportedModelIds>YETST</supportedModelIds>
    </supportedModelIds>
    <firmwareFilename>_</firmwareFilename>
    <firmwareVersion>_</firmwareVersion>
  </item>
  <item>
    <id>f0b7b35b-4b8e-4a15-9d66-91c4b3d575d1</id>
    <description>prav_Firm</description>
    <supportedModelIds>
      <supportedModelIds>PX013ANM</supportedModelIds>
      <supportedModelIds>PX013ANC</supportedModelIds>
    </supportedModelIds>
    <firmwareFilename>PX013AN_2.1s11_VBN_HYBse-signed.bin</firmwareFilename>
    <firmwareVersion>PX013AN_2.1s11_VBN_HYBse-signed</firmwareVersion>
  </item>
</ArrayList>
```

Create/update a firmware config

If firmware config is missing it will be being created, otherwise updated. For update operation id field is not needed.

POST http://<host>:<port>/updates/firmwares?applicationType={type}

Headers:

Content-Type = application/json OR application/xml

Accept = application/json OR application/xml

applicationType param is not required, default value is stb

Response: 200 OK and saved object (see get request for example); 400 BAD REQUEST (by validation error); 500 INTERNAL SERVER ERROR

Request:

<http://<host>:<port>/updates/firmwares.json>

OR

<http://<host>:<port>/updates/firmwares>

Restrictions:

Description, file name, version, supported model should be not empty;

JSON request

```
{
  "id": "b65962b5-1481-4eed-a010-2abfa8c3bbfd",
  "updated": 1440492963476,
  "description": "_-",
  "supportedModelIds": [
    "YETST"
  ],
  "firmwareDownloadProtocol": "tftp",
  "firmwareFilename": "_-",
  "firmwareVersion": "_-",
  "rebootImmediately": false
}
```

XML request

```
<FirmwareConfig>
  <id>b65962b5-1481-4eed-a010-2abfa8c3bbfd</id>
  <updated>1440492963476</updated>
  <description>_-</description>
  <supportedModelIds>
    <supportedModelIds>YETST</supportedModelIds>
  </supportedModelIds>
  <firmwareDownloadProtocol>tftp</firmwareDownloadProtocol>
  <firmwareFilename>_-</firmwareFilename>
  <firmwareVersion>_-</firmwareVersion>
  <rebootImmediately>false</rebootImmediately>
</FirmwareConfig>
```

Delete a firmware config by id

DELETE <http://<host>:<port>/delete/firmwares/<firmwareConfigId>>

Headers:

Accept = application/json OR application/xml

Response: 204 NO CONTENT and text message: Firmware config successfully deleted OR Config doesn't exist.

Request:

<http://<host>:<port>/delete/firmwares/b65962b5-1481-4eed-a010-2abfa8c3bbfd>

IP rules

Retrieve an ip rule list

GET <http://<host>:<port>/queries/rules/ips?applicationType={type}>

Headers:

Accept = application/json OR application/xml

Default value for applicationType parameter is stb

Response: 200 OK OR 400 BAD REQUEST (if application type has a wrong value)

Request:

<http://<host>:<port>/queries/rules/ips.json>

OR

<http://<host>:<port>/queries/rules/ips>

JSON response

```
[
  {
    "id": "ddc07355-d253-4f6b-8b42-296819d0d094",
    "name": "fsd",
    "ipAddressGroup": {
      "id": "2c184325-f9eb-4edc-85c3-5b6466fc3c5c",
      "name": "test",
      "ipAddresses": [
        "192.11.11.11"
      ]
    },
    "environmentId": "DEV",
    "modelId": "YETST",
    "noop": true,
    "expression": {
      "targetedModelIds": [
      ],
      "environmentId": "DEV",
      "modelId": "YETST",
      "ipAddressGroup": {
        "id": "2c184325-f9eb-4edc-85c3-5b6466fc3c5c",
        "name": "test",
        "ipAddresses": [
          "192.11.11.11"
        ]
      }
    }
  }
]
```

XML response


```

<ArrayList>
  <item>
    <id>ddc07355-d253-4f6b-8b42-296819d0d094</id>
    <name>fsd</name>
    <ipAddressGroup>
      <id>2c184325-f9eb-4edc-85c3-5b6466fc3c5c</id>
      <name>test</name>
      <ipAddresses>
        <ipAddresses>192.11.11.11</ipAddresses>
      </ipAddresses>
    </ipAddressGroup>
    <environmentId>DEV</environmentId>
    <modelId>YETST</modelId>
    <noop>true</noop>
    <expression>
      <targetedModelIds/> <environmentId>DEV</environmentId>
      <modelId>YETST</modelId>
      <ipAddressGroup>
        <id>2c184325-f9eb-4edc-85c3-5b6466fc3c5c</id>
        <name>test</name>
        <ipAddresses>
          <ipAddresses>192.11.11.11</ipAddresses>
        </ipAddresses>
      </ipAddressGroup>
    </expression>
  </item>
</ArrayList>

```

Retrieve an ip rule by name

GET <http://<host>:<port>/queries/rules/ips/{ipRuleName}?applicationType={type}>

Headers:

Accept = application/json OR application/xml

Default value for applicationType parameter is stb

Response: 200 OK OR 400 BAD REQUEST (if application type has a wrong value)

Request:

<http://<host>:<port>/queries/rules/ips/fsd.json>

OR

<http://<host>:<port>/queries/rules/ips/fsd>

JSON response

```
{
  "id": "ddc07355-d253-4f6b-8b42-296819d0d094",
  "name": "fsd",
  "ipAddressGroup": {
    "id": "2c184325-f9eb-4edc-85c3-5b6466fc3c5c",
    "name": "test",
    "ipAddresses": [
      "192.11.11.11"
    ]
  },
  "environmentId": "DEV",
  "modelId": "YETST",
  "noop": true,
  "expression": {
    "targetedModelIds": [

    ],
    "environmentId": "DEV",
    "modelId": "YETST",
    "ipAddressGroup": {
      "id": "2c184325-f9eb-4edc-85c3-5b6466fc3c5c",
      "name": "test",
      "ipAddresses": [
        "192.11.11.11"
      ]
    }
  }
}
```

XML response

```

<IpRuleBean>
  <id>ddc07355-d253-4f6b-8b42-296819d0d094</id>
  <name>fsd</name>
  <ipAddressGroup>
    <id>2c184325-f9eb-4edc-85c3-5b6466fc3c5c</id>
    <name>test</name>
    <ipAddresses>
      <ipAddresses>192.11.11.11</ipAddresses>
    </ipAddresses>
  </ipAddressGroup>
  <environmentId>DEV</environmentId>
  <modelId>YETST</modelId>
  <noop>true</noop>
  <expression>
    <targetedModelIds/><environmentId>DEV</environmentId>
    <modelId>YETST</modelId>
    <ipAddressGroup>
      <id>2c184325-f9eb-4edc-85c3-5b6466fc3c5c</id>
      <name>test</name>
      <ipAddresses>
        <ipAddresses>192.11.11.11</ipAddresses>
      </ipAddresses>
    </ipAddressGroup>
  </expression>
</IpRuleBean>

```

Create/update an ip rule

If IpRule is missing it will be being created, otherwise updated. For update operation id field is not needed.

POST <http://<host>:<port>/updates/rules/ips?applicationType={type}>

Headers:

Content-Type = application/json OR application/xml

Accept = application/json OR application/xml

Response: 200 OK and saved object (see get request for example); 400 BAD REQUEST; 500 INTERNAL SERVER ERROR

Restrictions:

Name, environmentId, modelId should be not empty, IP address group should be specified

Request:

<http://<host>:<port>/updates/rules/ips.json>

OR

<http://<host>:<port>/updates/rules/ips>

JSON request

```
{
  "id": "ddc07355-d253-4f6b-8b42-296819d0d094",
  "name": "fsd",
  "ipAddressGroup": {
    "id": "2c184325-f9eb-4edc-85c3-5b6466fc3c5c",
    "name": "test",
    "ipAddresses": [
      "192.11.11.11"
    ]
  },
  "environmentId": "DEV",
  "modelId": "YETST",
  "noop": true,
  "expression": {
    "targetedModelIds": [
    ],
    "environmentId": "DEV",
    "modelId": "YETST",
    "ipAddressGroup": {
      "id": "2c184325-f9eb-4edc-85c3-5b6466fc3c5c",
      "name": "test",
      "ipAddresses": [
        "192.11.11.11"
      ]
    }
  }
}
```

XML request

```

<IpRuleBean>
  <id>ddc07355-d253-4f6b-8b42-296819d0d094</id>
  <name>fsd</name>
  <ipAddressGroup>
    <id>2c184325-f9eb-4edc-85c3-5b6466fc3c5c</id>
    <name>test</name>
    <ipAddresses>
      <ipAddresses>192.11.11.11</ipAddresses>
    </ipAddresses>
  </ipAddressGroup>
  <environmentId>DEV</environmentId>
  <modelId>YETST</modelId>
  <noop>true</noop>
  <expression>
    <targetedModelIds/><environmentId>DEV</environmentId>
    <modelId>YETST</modelId>
    <ipAddressGroup>
      <id>2c184325-f9eb-4edc-85c3-5b6466fc3c5c</id>
      <name>test</name>
      <ipAddresses>
        <ipAddresses>192.11.11.11</ipAddresses>
      </ipAddresses>
    </ipAddressGroup>
  </expression>
</IpRuleBean>

```

Delete an ip rule

DELETE <http://<host>:<port>/delete/rules/ips/{ipRuleName}?applicationType={type}>

Headers:

Accept = application/json OR application/xml

applicationType parameter is not required, default value is stb

Response: 204 NO CONTENT and message: IpRule successfully deleted OR Rule doesn't exists, 400 BAD REQUEST if applicationType is not valid;

Request: <http://<host>:<port>/delete/rules/ips/ruleName>

Location filter

Retrieve a location filter list

GET <http://<host>:<port>/queries/filters/locations?applicationType={type}>

Headers:

Accept = application/json OR application/xml

applicationType parameter is not required, default value is stb

Response: 200 OK OR 400 BAD REQUEST if applicationType is not valid

Request:

<http://<host>:<port>/queries/filters/locations.json>

OR

<http://<host>:<port>/queries/filters/locations>

JSON response

```
[
  {
    "ipAddressGroup": {
      "id": "2c184325-f9eb-4edc-85c3-5b6466fc3c5c",
      "name": "test",
      "ipAddresses": [
        "192.11.11.11"
      ]
    },
    "environments": [

    ],
    "models": [

    ],
    "ipv6FirmwareLocation": "2001:0db8:11a3:09d7:1f34:8a2e:07a0:765d",
    "httpLocation": "http://localhost:8080",
    "forceHttp": true,
    "id": "2ce1279b-bb25-4fda-9a34-fe8466bc2702",
    "name": "name",
    "boundConfigId": "95e75859-ae8f-4d6a-b758-11fefbe647e1",
    "ipv4FirmwareLocation": "10.10.10.10"
  }
]
```

XML response

```
<DownloadLocationFilterWrapper>
  <ipAddressGroup>
    <id>2c184325-f9eb-4edc-85c3-5b6466fc3c5c</id>
    <name>test</name>
    <ipAddresses>
      <ipAddresses>192.11.11.11</ipAddresses>
    </ipAddresses>
  </ipAddressGroup>
</environments/>
<models/>
<ipv6FirmwareLocation>2001:0db8:11a3:09d7:1f34:8a2e:07a0:765d</ipv6FirmwareLocation>
<httpLocation>http://localhost:8080</httpLocation>
<forceHttp>true</forceHttp>
<id>2ce1279b-bb25-4fda-9a34-fe8466bc2702</id>
<name>name</name>
<boundConfigId>95e75859-ae8f-4d6a-b758-11fefbe647e1</boundConfigId>
<ipv4FirmwareLocation>10.10.10.10</ipv4FirmwareLocation>
</DownloadLocationFilterWrapper>
```

Retrieve a location filter by name

GET <http://<host>:<port>/queries/filters/locations/{locationFilterName}?applicationType={type}>

or legacy endpoint

GET <http://<host>:<port>/queries/filters/locations/byName/{locationFilterName}>

Headers:

Accept = application/json OR application/xml

Response: 200 OK

Request:

<http://<host>:<port>/queries/filters/locations/name.json>

OR

<http://<host>:<port>/queries/filters/locations/name>

JSON response

```
[
  {
    "ipAddressGroup": {
      "id": "2c184325-f9eb-4edc-85c3-5b6466fc3c5c",
      "name": "test",
      "ipAddresses": [
        "192.11.11.11"
      ]
    },
    "environments": [

    ],
    "models": [

    ],
    "ipv6FirmwareLocation": "2001:0db8:11a3:09d7:1f34:8a2e:07a0:765d",
    "httpLocation": "http://localhost:8080",
    "forceHttp": true,
    "id": "2ce1279b-bb25-4fda-9a34-fe8466bc2702",
    "name": "name",
    "boundConfigId": "95e75859-ae8f-4d6a-b758-11fefbe647e1",
    "ipv4FirmwareLocation": "10.10.10.10"
  }
]
```

XML response


```

<DownloadLocationFilterWrapper>
  <ipAddressGroup>
    <id>2c184325-f9eb-4edc-85c3-5b6466fc3c5c</id>
    <name>test</name>
    <ipAddresses>
      <ipAddresses>192.11.11.11</ipAddresses>
    </ipAddresses>
  </ipAddressGroup>
  <environments/>
  <models/>
  <ipv6FirmwareLocation>2001:0db8:11a3:09d7:1f34:8a2e:07a0:765d</ipv6FirmwareLocation>
  <httpLocation>http://localhost:8080</httpLocation>
  <forceHttp>true</forceHttp>
  <id>2ce1279b-bb25-4fda-9a34-fe8466bc2702</id>
  <name>name</name>
  <boundConfigId>95e75859-ae8f-4d6a-b758-11fefbe647e1</boundConfigId>
  <ipv4FirmwareLocation>10.10.10.10</ipv4FirmwareLocation>
</DownloadLocationFilterWrapper>

```

Create/update location filter

If location filter is missing it will be being created, otherwise updated. For update operation id field is not needed.

POST <http://<host>:<port>/updates/filters/locations?applicationType={type}>

Headers:

Content-Type = application/json OR application/xml

Accept = application/json OR application/xml

applicationType parameter is not required, default value is stb

Response: 200 OK and saved object (see get request for example); 400 BAD REQUEST; 500 INTERNAL SERVER ERROR

Restrictions:

Condition, models, environments, IPv4, location, any location (HTTP or firmware), IPv4/IPv6 should be valid

Request:

<http://<host>:<port>/updates/filters/locations.json>

OR

<http://<host>:<port>/updates/filters/locations>

XML request

```

<DownloadLocationFilterWrapper>
  <ipAddressGroup>
    <id>2c184325-f9eb-4edc-85c3-5b6466fc3c5c</id>
    <name>test</name>
    <ipAddresses>
      <ipAddresses>192.11.11.11</ipAddresses>
    </ipAddresses>
  </ipAddressGroup>
  <environments/>
  <models/>
  <ipv6FirmwareLocation>2001:0db8:11a3:09d7:1f34:8a2e:07a0:765d</ipv6FirmwareLocation>
  <httpLocation>http://localhost:8080</httpLocation>
  <forceHttp>true</forceHttp>
  <id>2ce1279b-bb25-4fda-9a34-fe8466bc2702</id>
  <name>name</name>
  <boundConfigId>95e75859-ae8f-4d6a-b758-11fefbe647e1</boundConfigId>
  <ipv4FirmwareLocation>10.10.10.10</ipv4FirmwareLocation>
</DownloadLocationFilterWrapper>

```

JSON request

```
{
  "ipAddressGroup": {
    "id": "2c184325-f9eb-4edc-85c3-5b6466fc3c5c",
    "name": "test",
    "ipAddresses": [
      "192.11.11.11"
    ]
  },
  "environments": [

  ],
  "models": [

  ],
  "ipv6FirmwareLocation": "2001:0db8:11a3:09d7:1f34:8a2e:07a0:765d",
  "httpLocation": "http://localhost:8080",
  "forceHttp": true,
  "id": "2ce1279b-bb25-4fda-9a34-fe8466bc2702",
  "name": "name",
  "boundConfigId": "95e75859-ae8f-4d6a-b758-11fefbe647e1",
  "ipv4FirmwareLocation": "10.10.10.10"
}
```

Delete location filter by name

DELETE <http://<host>:<port>/delete/filters/locations/{locationFilterName}?applicationType={type}>

Headers:

Accept = application/json OR application/xml

Response: 204 NO CONTENT and message: Location filter successfully deleted OR Filter doesn't exist with name: <filterName>, 400 BAD REQUEST if applicationType is not valid

Request: <http://<host>:<port>/delete/filters/location/name>

Download location filter

Retrieve download location filter

GET <http://<host>:<port>/queries/filters/downloadlocation?applicationType={type}>

Headers:

Accept = application/json OR application/xml

applicationType param is not required, default value is stb

Response: 200 OK, 400 BAD REQUEST if applicationType is not valid

Request:

<http://<host>:<port>/queries/filters/downloadlocation.json>

OR

<http://<host>:<port>/queries/filters/downloadlocation>

JSON response

```
{
  "type": "com.comcast.xconf.estbfirmware.DownloadLocationRoundRobinFilterValue",
  "id": "DOWNLOAD_LOCATION_ROUND_ROBIN_FILTER_VALUE",
  "locations": [
    {
      "locationIp": "10.10.10.10",
      "percentage": 100.0
    }
  ],
  "ipv6locations": [

  ],
  "rogueModels": [

  ],
  "httpLocation": "lf.com",
  "httpFullUrlLocation": "http://www.localhost.org",
  "neverUseHttp": true,
  "firmwareVersions": "??"
}
```

XML response

```
<DownloadLocationRoundRobinFilterValue type="com.comcast.xconf.estbfirmware.
DownloadLocationRoundRobinFilterValue">
  <id>DOWNLOAD_LOCATION_ROUND_ROBIN_FILTER_VALUE</id>
  <locations>
    <locations>
      <locationIp>10.10.10.10</locationIp>
      <percentage>100.0</percentage>
    </locations>
  </locations>
  <ipv6locations/>
  <rogueModels/>
  <httpLocation>lf.com</httpLocation>
  <httpFullUrlLocation>http://www.localhost.org</httpFullUrlLocation>
  <neverUseHttp>true</neverUseHttp>
  <firmwareVersions>??</firmwareVersions>
</DownloadLocationRoundRobinFilterValue>
```

Update download location filter

POST <http://<host>:<port>/updates/filters/downloadlocation?applicationType={type}>

Headers:

Content-Type = application/json OR application/xml

Accept = application/json OR application/xml

applicationType param is not required, default value is stb

Response: 200 OK and saved object (see get request); 400 BAD REQUEST; 500 INTERNAL SERVER ERROR

Restrictions:

Location URL, IPv4/IPv6 should be valid, percentage should be positive and within [0, 100], locations should be not duplicated

Request:

<http://<host>:<port>/updates/filters/downloadlocation.json>

OR

<http://<host>:<port>/updates/filters/downloadlocation.json>

JSON request

```

{
  "type": "com.comcast.xconf.estbfirmware.DownloadLocationRoundRobinFilterValue",
  "id": "DOWNLOAD_LOCATION_ROUND_ROBIN_FILTER_VALUE",
  "updated": 1441287139144,
  "locations": [
    {
      "locationIp": "10.10.10.10",
      "percentage": 100.0
    }
  ],
  "ipv6locations": [
  ],
  "rogueModels": [
  ],
  "httpLocation": "lf.com",
  "httpFullUrlLocation": "http://www.localhost.org",
  "neverUseHttp": true,
  "firmwareVersions": "??"
}

```

XML request

```

<DownloadLocationRoundRobinFilterValue type="com.comcast.xconf.estbfirmware.
DownloadLocationRoundRobinFilterValue">
  <id>DOWNLOAD_LOCATION_ROUND_ROBIN_FILTER_VALUE</id>
  <updated>1441287139144</updated>
  <locations>
    <locations>
      <locationIp>10.10.10.10</locationIp>
      <percentage>100.0</percentage>
    </locations>
  </locations>
  <ipv6locations/>
  <rogueModels/>
  <httpLocation>lf.com</httpLocation>
  <httpFullUrlLocation>http://www.localhost.org</httpFullUrlLocation>
  <neverUseHttp>true</neverUseHttp>
  <firmwareVersions>??</firmwareVersions>
</DownloadLocationRoundRobinFilterValue>

```

Environment model rules

Retrieve an environment model rule list

GET <http://<host>:<port>/queries/rules/envModels?applicationType={type}>

Headers:

Accept = application/json OR application/xml

applicationType param is not required, default value is stb

Response: 200 OK, 400 BAD REQUEST if applicationType is not valid

Request:

<http://<host>:<port>/queries/rules/envModels.json>

OR

<http://<host>:<port>/queries/rules/envModels>

JSON response

```
[
  {
    "id": "12b620bd-2e74-4467-91e5-c29657022c05",
    "name": "re",
    "firmwareConfig": {
      "id": "f0b7b35b-4b8e-4a15-9d66-91c4b3d575d1",
      "description": "prav_Firm",
      "supportedModelIds": [
        "PX013ANM",
        "PX013ANC"
      ],
      "firmwareFilename": "PX013AN_2.1s11_VBN_HYBse-signed.bin",
      "firmwareVersion": "PX013AN_2.1s11_VBN_HYBse-signed"
    },
    "environmentId": "TEST",
    "modelId": "PX013ANC"
  },
  {
    "id": "3609a195-ecd3-4950-8583-ceb2f6f6f16b",
    "name": "uuuuuuu",
    "environmentId": "DEV",
    "modelId": "YETST"
  }
]
```

XML response

```
<ArrayList>
  <item>
    <id>12b620bd-2e74-4467-91e5-c29657022c05</id>
    <name>re</name>
    <firmwareConfig>
      <id>f0b7b35b-4b8e-4a15-9d66-91c4b3d575d1</id>
      <description>prav_Firm</description>
      <supportedModelIds>
        <supportedModelIds>PX013ANM</supportedModelIds>
        <supportedModelIds>PX013ANC</supportedModelIds>
      </supportedModelIds>
      <firmwareFilename>PX013AN_2.1s11_VBN_HYBse-signed.bin</firmwareFilename>
      <firmwareVersion>PX013AN_2.1s11_VBN_HYBse-signed</firmwareVersion>
    </firmwareConfig>
    <environmentId>TEST</environmentId>
    <modelId>PX013ANC</modelId>
  </item>
  <item>
    <id>3609a195-ecd3-4950-8583-ceb2f6f6f16b</id>
    <name>uuuuuuu</name>
    <environmentId>DEV</environmentId>
    <modelId>YETST</modelId>
  </item>
</ArrayList>
```

Retrieve an environment model rule by name

GET <http://<host>:<port>/queries/rules/envModels/{envModelRuleName}?applicationType={type}>

Headers:

Accept = application/json OR application/xml

Response: 200 OK, 400 BAD REQUEST if applicationType is not valid

Request:

<http://<host>:<port>/queries/rules/envModels/testName.json>

OR

<http://<host>:<port>/queries/rules/envModels/testName.json>

JSON response

```
{
  "id": "12b620bd-2e74-4467-91e5-c29657022c05",
  "name": "testName",
  "firmwareConfig": {
    "id": "f0b7b35b-4b8e-4a15-9d66-91c4b3d575d1",
    "description": "prav_Firm",
    "supportedModelIds": [
      "PX013ANM",
      "PX013ANC"
    ],
    "firmwareFilename": "PX013AN_2.1s11_VBN_HYBse-signed.bin",
    "firmwareVersion": "PX013AN_2.1s11_VBN_HYBse-signed"
  },
  "environmentId": "TEST",
  "modelId": "PX013ANC"
}
```

XML response

```
<EnvModelRuleBean>
  <id>12b620bd-2e74-4467-91e5-c29657022c05</id>
  <name>testName</name>
  <firmwareConfig>
    <id>f0b7b35b-4b8e-4a15-9d66-91c4b3d575d1</id>
    <description>prav_Firm</description>
    <supportedModelIds>
      <supportedModelIds>PX013ANM</supportedModelIds>
      <supportedModelIds>PX013ANC</supportedModelIds>
    </supportedModelIds>
    <firmwareFilename>PX013AN_2.1s11_VBN_HYBse-signed.bin</firmwareFilename>
    <firmwareVersion>PX013AN_2.1s11_VBN_HYBse-signed</firmwareVersion>
  </firmwareConfig>
  <environmentId>TEST</environmentId>
  <modelId>PX013ANC</modelId>
</EnvModelRuleBean>
```

Create/update an environment model rule

If EnvModelRule is missing it will be being created, otherwise updated. For update operation id field is not needed.

POST <http://<host>:<port>/updates/rules/envModels?applicationType={type}>

Headers:

Content-Type = application/json OR application/xml

Accept = application/json OR application/xml

applicationType param is not required, default value is stb

Response: 200 OK and saved object (see get request); 400 BAD REQUEST; 500 INTERNAL SERVER ERROR

Restrictions:

Name, environment, model should be not empty, name is used only once, env/model should not overlap each other

Request:

<http://<host>:<port>/updates/rules/envModels.json>

OR

<http://<host>:<port>/updates/rules/envModels>

JSON request

```
{
  "id": "12b620bd-2e74-4467-91e5-c29657022c05",
  "name": "testName",
  "firmwareConfig": {
    "id": "f0b7b35b-4b8e-4a15-9d66-91c4b3d575d1",
    "description": "prav_Firm",
    "supportedModelIds": [
      "PX013ANM",
      "PX013ANC"
    ],
    "firmwareFilename": "PX013AN_2.1s11_VBN_HYBse-signed.bin",
    "firmwareVersion": "PX013AN_2.1s11_VBN_HYBse-signed"
  },
  "environmentId": "TEST",
  "modelId": "PX013ANC"
}
```

XML request

```
<EnvModelRuleBean>
  <id>12b620bd-2e74-4467-91e5-c29657022c05</id>
  <name>testName</name>
  <firmwareConfig>
    <id>f0b7b35b-4b8e-4a15-9d66-91c4b3d575d1</id>
    <description>prav_Firm</description>
    <supportedModelIds>
      <supportedModelIds>PX013ANM</supportedModelIds>
      <supportedModelIds>PX013ANC</supportedModelIds>
    </supportedModelIds>
    <firmwareFilename>PX013AN_2.1s11_VBN_HYBse-signed.bin</firmwareFilename>
    <firmwareVersion>PX013AN_2.1s11_VBN_HYBse-signed</firmwareVersion>
  </firmwareConfig>
  <environmentId>TEST</environmentId>
  <modelId>PX013ANC</modelId>
</EnvModelRuleBean>
```

Delete an environment model rule

DELETE <http://<host>:<port>/delete/rules/envModels/{envModelRuleName}?applicationType={type}>

Headers:

Accept = application/json OR application/xml

applicationType param is not required, default value is stb

Response: 204 NO CONTENT and message: Rule successfully deleted OR Rule doesn't exist with name: <ruleName>, 400 if applicationType is not valid

Request: <http://<host>:<port>/delete/rules/envModels/testName.json>

IP filter

IP Filter

Retrieve an IP filter list

GET <http://<host>:<port>/queries/filters/ips?applicationType={type}>

Headers:

Accept = application/json OR application/xml

applicationType param is not required, default value is stb

Response: 200 OK, 400 BAD REQUEST

Request:

<http://<host>:<port>/queries/filters/ips.json>

OR

<http://<host>:<port>/queries/filters/ips>

JSON response

```
[
  {
    "id": "8bdb3493-a18b-4230-9b25-fd44df38863b",
    "name": "name",
    "ipAddressGroup": {
      "id": "2c184325-f9eb-4edc-85c3-5b6466fc3c5c",
      "name": "test",
      "ipAddresses": [
        "192.11.11.11"
      ]
    },
    "warehouse": false
  },
  {
    "id": "f9c5a6e8-d34f-4dc6-ae41-9016b70552ae",
    "name": "namef",
    "ipAddressGroup": {
      "id": "2c184325-f9eb-4edc-85c3-5b6466fc3c5c",
      "name": "test",
      "ipAddresses": [
        "192.11.11.11"
      ]
    },
    "warehouse": false
  }
]
```

XML response

```
<ArrayList>
  <item>
    <id>8bdb3493-a18b-4230-9b25-fd44df38863b</id>
    <name>name</name>
    <ipAddressGroup>
      <id>2c184325-f9eb-4edc-85c3-5b6466fc3c5c</id>
      <name>test</name>
      <ipAddresses>
        <ipAddresses>192.11.11.11</ipAddresses>
      </ipAddresses>
    </ipAddressGroup>
    <warehouse>false</warehouse>
  </item>
  <item>
    <id>f9c5a6e8-d34f-4dc6-ae41-9016b70552ae</id>
    <name>namef</name>
    <ipAddressGroup>
      <id>2c184325-f9eb-4edc-85c3-5b6466fc3c5c</id>
      <name>test</name>
      <ipAddresses>
        <ipAddresses>192.11.11.11</ipAddresses>
      </ipAddresses>
    </ipAddressGroup>
    <warehouse>false</warehouse>
  </item>
</ArrayList>
```

Retrieve an ip filter by name

GET <http://<host>:<port>/queries/filters/ips/{ipFilterName}?applicationType={type}>

Headers:

Accept = application/json OR application/xml

applicationType param is not required, default value is stb

Response: 200 OK, 400 BAD REQUEST if applicationType is not valid

Request:

<http://<host>:<port>/queries/filters/ips/namef.json>

OR

<http://<host>:<port>/queries/filters/ips/namef.json>

JSON response

```
{
  "id": "f9c5a6e8-d34f-4dc6-ae41-9016b70552ae",
  "name": "namef",
  "ipAddressGroup": {
    "id": "2c184325-f9eb-4edc-85c3-5b6466fc3c5c",
    "name": "test",
    "ipAddresses": [
      "192.11.11.11"
    ]
  },
  "warehouse": false
}
```

XML response

```
<IpFilter>
  <id>8bdb3493-a18b-4230-9b25-fd44df38863b</id>
  <name>name</name>
  <ipAddressGroup>
    <id>2c184325-f9eb-4edc-85c3-5b6466fc3c5c</id>
    <name>test</name>
    <ipAddresses>
      <ipAddresses>192.11.11.11</ipAddresses>
    </ipAddresses>
  </ipAddressGroup>
  <warehouse>false</warehouse>
</IpFilter>
```

Create/update an IP filter

If IpFilter is missing it will be being created, otherwise updated. For update operation id field is not needed.

POST <http://<host>:<port>/updates/filters/ips?applicationType={type}>

Headers:

Content-Type = application/json OR application/xml

Accept = application/json OR application/xml

applicationType param is not required, default value is stb

Response: 200 OK and saved object (see get request); 400 BAD REQUEST; 500 INTERNAL SERVER ERROR

Restrictions:

Name, IP address group should be not empty

Request:

<http://<host>:<port>/updates/filters/ips.json>

OR

<http://<host>:<port>/updates/filters/ips>

JSON request

```
{
  "id": "f9c5a6e8-d34f-4dc6-ae41-9016b70552ae",
  "name": "namef",
  "ipAddressGroup": {
    "id": "2c184325-f9eb-4edc-85c3-5b6466fc3c5c",
    "name": "test",
    "ipAddresses": [
      "192.11.11.11"
    ]
  },
  "warehouse": false
}
```

XML request

```
<IpFilter>
  <id>8bdbb3493-a18b-4230-9b25-fd44df38863b</id>
  <name>name</name>
  <ipAddressGroup>
    <id>2c184325-f9eb-4edc-85c3-5b6466fc3c5c</id>
    <name>test</name>
    <ipAddresses>
      <ipAddresses>192.11.11.11</ipAddresses>
    </ipAddresses>
  </ipAddressGroup>
  <warehouse>false</warehouse>
</IpFilter>
```

Delete IP filter

Response: 200 OK and message: IpRuleId is empty OR IpFilter successfully deleted

DELETE <http://<host>:<port>/delete/filters/ips/{ipFilterName}?applicationType={stb}>

Headers:

Accept = application/json OR application/xml

applicationType param is not required, default value is stb

Response: 204 NO CONTENT and message: IpFilter successfully deleted OR Filter doesn't exist with name: <filterName>, 400 BAD REQUEST if applicationType is not valid

Request: <http://<host>:<port>/delete/filters/ips/namef>

Percent filter

Retrieve percent filter

GET <http://<host>:<port>/queries/filters/percent?applicationType={type}>

Headers:

Accept = application/json OR application/xml

applicationType param is not required, default value is stb

Response: 200 OK, 400 BAD REQUEST if applicationType is not valid

Request:

<http://<host>:<port>/queries/filters/percent.json>

OR

<http://<host>:<port>/queries/filters/percent>

Note

firmwareVersions will be present only if firmwareVersions is set to true, lastKnownGood and intermediateVersion will be set to empty string and empty tag in the JSON and XML response respectively if they are not set in the Percent Filter.

JSON response

```

{
  "type": "com.comcast.xconf.queries.beans.PercentFilterWrapper",
  "id": "PERCENT_FILTER_VALUE",
  "whitelist":
  {
    "id": "66e77146-2542-4173-99fb-2acfe52b9c06",
    "name": "All warehouses",
    "ipAddresses":
    [
      "2345678", "76.20.128.0/21",
      "76.106.43.0/24",
      "", "162.150.244.212"
    ]
  },
  "percentage": 100.0,
  "EnvModelPercentages":
  [
    {
      "percentage": 100.0,
      "active": true,
      "firmwareCheckRequired": true,
      "rebootImmediately": true,
      "lastKnownGood": "", "intermediateVersion": "",
      "firmwareVersions":
      [
        "addtestedit"
      ],
      "name": "!@"
    },
    {
      "percentage": 50.0,
      "active": true,
      "firmwareCheckRequired": true,
      "rebootImmediately": true,
      "lastKnownGood": "PR150BN_2.4p6s3_VBN_HYBsd",
      "intermediateVersion": "PR150BN_2.3p7s1_PROD_HYBse",
      "firmwareVersions":
      [
        "PR150BN_2.4p6s3_VBN_HYBsd",
        "PR150BN_2.4p8s3_VBN_HYBsd-signed",
        "PR150BN_2.3p7s1_PROD_HYBse"
      ],
      "name": "IV_6a"
    }
  ]
}

```

XML response


```

<PercentFilterWrapper type="com.comcast.xconf.queries.beans.PercentFilterWrapper">
  <id>PERCENT_FILTER_VALUE</id>
  <percentage>99.0</percentage>
  <EnvModelPercentages>
    <percentage>100.0</percentage>
    <active>false</active>
    <firmwareCheckRequired>false</firmwareCheckRequired>
    <rebootImmediately>false</rebootImmediately>
    <lastKnownGood/>
    <intermediateVersion/>
    <firmwareVersions/>
    <name>123</name>
  </EnvModelPercentages>
  <EnvModelPercentages>
    <percentage>50.0</percentage>
    <active>true</active>
    <firmwareCheckRequired>true</firmwareCheckRequired>
    <rebootImmediately>true</rebootImmediately>
    <lastKnownGood>PR150BN_2.4p6s3_VBN_HYBsd</lastKnownGood>
    <intermediateVersion>PR150BN_2.3p7s1_PROD_HYBse</intermediateVersion>
    <firmwareVersions>
      <firmwareVersions>PR150BN_2.4p6s3_VBN_HYBsd</firmwareVersions>
      <firmwareVersions>PR150BN_2.4p8s3_VBN_HYBsd-signed</firmwareVersions>
      <firmwareVersions>PR150BN_2.3p7s1_PROD_HYBse</firmwareVersions>
    </firmwareVersions>
    <name>IV_6a</name>
  </EnvModelPercentages>
</PercentFilterWrapper>

```

Retrieve percent filter field values

GET <http://<host>:<port>/queries/filters/percent?field=fieldName&applicationType={type}>

Headers:

Accept = application/json OR application/xml

applicationType param is not required, default value is stb

Response: 200 OK, 400 BAD REQUEST if applicationType is not valid

Request: <http://<host>:<port>/queries/filters/percent?field=fieldName>

if field does not exists:

Response: 404 Not Found and Message: Field fieldName does not exist

JSON response

```

{
  "firmwareVersions":
  [
    "firmwareVersion1", "firmwareVersion2"
  ]
}

```

XML response

```

<SingletonMap>
  <fieldName>value1</fieldName>
  <fieldName>value2</fieldName>
</SingletonMap>

```

Update percent filter

POST <http://<host>:<port>/updates/filters/percent?applicationType={type}>

Headers:

Content-Type = application/json OR application/xml

Accept = application/json OR application/xml

applicationType param is not required, default value is stb

Response: 200 OK and saved object (see get request); 400 BAD REQUEST; 500 INTERNAL SERVER ERROR

Restrictions:

Percentage should be positive and within [0, 100],

Request:

<http://<host>:<port>/updates/filters/percent.json>

OR

<http://<host>:<port>/updates/filters/percent>

JSON request

```
{
  "type": "com.comcast.xconf.queries.beans.PercentFilterWrapper",
  "id": "PERCENT_FILTER_VALUE",
  "percentage": 99.0,
  "EnvModelPercentages": [
    {
      "percentage": 100.0,
      "active": false,
      "firmwareCheckRequired": false,
      "rebootImmediately": false,
      "firmwareVersions": [

      ],
      "name": "re"
    },
    {
      "percentage": 100.0,
      "active": false,
      "firmwareCheckRequired": false,
      "rebootImmediately": false,
      "firmwareVersions": [

      ],
      "name": "uuuuuuu"
    }
  ]
}
```

XML request

```
<PercentFilterWrapper type="com.comcast.xconf.queries.beans.PercentFilterWrapper">
  <id>PERCENT_FILTER_VALUE</id>
  <percentage>99.0</percentage>
  <EnvModelPercentages>
    <EnvModelPercentages>
      <percentage>100.0</percentage>
      <active>false</active>
      <firmwareCheckRequired>false</firmwareCheckRequired>
      <rebootImmediately>false</rebootImmediately>
      <firmwareVersions/>
      <name>re</name>
    </EnvModelPercentages>
  </EnvModelPercentages>
  <percentage>100.0</percentage>
  <active>false</active>
  <firmwareCheckRequired>false</firmwareCheckRequired>
  <rebootImmediately>false</rebootImmediately>
  <firmwareVersions/>
  <name>uuuuuuu</name>
</EnvModelPercentages>
</PercentFilterWrapper>
```

Retrive EnvModelPercentages

GET <http://<host>:<port>/queries/percentageBean?applicationType={type}>

Headers:

Content-Type = application/json OR application/xml

Accept = application/json OR application/xml

applicationType param is not required, default value is stb

Response: 200 OK, 400 BAD REQUEST

Request: <http://<host>:<port>/queries/percentageBean>

JSON response

```
[
  {
    "id": "4c2cea9e-c1be-4219-8f2a-70084f1cb648",
    "name": "Percent Test 2",
    "active": false,
    "firmwareCheckRequired": false,
    "rebootImmediately": false,
    "firmwareVersions": [],
    "environment": "envId",
    "model": "modelId",
    "distributions": [
      {
        "configId": "0c176c01-b4cf-4706-9e41-5118e30f68cb",
        "percentage": 100.0
      }
    ]
  }
]
```

XML response

```
<ArrayList>
  <item>
    <id>4c2cea9e-c1be-4219-8f2a-70084f1cb648</id>
    <name>Percent Test 2</name>
    <active>false</active>
    <firmwareCheckRequired>false</firmwareCheckRequired>
    <rebootImmediately>false</rebootImmediately>
    <firmwareVersions/>
    <environment>envId</environment>
    <model>modelId</model>
    <distributions>
      <distributions>
        <configId>29ab0494-0ee9-406a-9189-a81598988a54</configId>
        <percentage>100.0</percentage>
      </distributions>
    </distributions>
  </item>
</ArrayList>
```

Retrive EnvModelPercentage by id

GET <http://<host>:<port>/queries/percentageBean/id>

Headers:

Content-Type = application/json OR application/xml

Accept = application/json OR application/xml

Response: 200 OK OR 404 if envModelPercentage is not found

Request: <http://<host>:<port>/queries/percentageBean/4c2cea9e-c1be-4219-8f2a-70084f1cb648>

JSON response

```
{
  "id": "4c2cea9e-c1be-4219-8f2a-70084f1cb648",
  "name": "Percent Test 2",
  "active": false,
  "firmwareCheckRequired": false,
  "rebootImmediately": false,
  "firmwareVersions": [],
  "environment": "envId",
  "model": "modelId",
  "distributions": [
    {
      "configId": "0c176c01-b4cf-4706-9e41-5118e30f68cb",
      "percentage": 100.0
    }
  ]
}
```

XML response

```
<PercentageBean>
  <id>4c2cea9e-c1be-4219-8f2a-70084f1cb648</id>
  <ttlMap/>
  <name>Percent Test 2</name>
  <active>false</active>
  <firmwareCheckRequired>false</firmwareCheckRequired>
  <rebootImmediately>false</rebootImmediately>
  <firmwareVersions/>
  <environment>envId</environment>
  <model>modelId</model>
  <distributions>
    <distributions>
      <configId>0c176c01-b4cf-4706-9e41-5118e30f68cb</configId>
      <percentage>100.0</percentage>
    </distributions>
  </distributions>
</PercentageBean>
```

Create envModelPercentage

POST <http://<host>:<port>/updates/percentageBean?applicationType={type}>

Headers:

Content-Type = application/json OR application/xml

Accept = application/json OR application/xml

applicationType param is not valid, default value is stb

Response: 200 OK, 404 NOT FOUND, 409 CONFLICT, 400 BAD REQUEST

Request: <http://<host>:<port>/queries/percentageBean>

Restrictions:

Name should be unique and not blank, environment and model should be not empty, at least one firmware version should be in minCheck list if firmwareCheckRequired=true, percentage within [0, 100], distribution firmware version should be in minCheck list if firmwareCheckRequired=true, total distribution percentage is within [0, 100], last known good is not empty if total distribution percentage < 100, last known good should be selected in minCheck list.

JSON request

```
{
  "id": "4c2cea9e-c1be-4219-8f2a-70084f1cb648",
  "name": "Percent Test 2",
  "active": false,
  "firmwareCheckRequired": false,
  "rebootImmediately": false,
  "firmwareVersions": [],
  "environment": "envId",
  "model": "modelId",
  "distributions": [
    {
      "configId": "0c176c01-b4cf-4706-9e41-5118e30f68cb",
      "percentage": 100.0
    }
  ]
}
```

XML request

```
<PercentageBean>
  <id>4c2cea9e-c1be-4219-8f2a-70084f1cb648</id>
  <ttlMap/>
  <name>Percent Test 2</name>
  <active>false</active>
  <firmwareCheckRequired>false</firmwareCheckRequired>
  <rebootImmediately>false</rebootImmediately>
  <firmwareVersions/>
  <environment>envId</environment>
  <model>modelId</model>
  <distributions>
    <distributions>
      <configId>0c176c01-b4cf-4706-9e41-5118e30f68cb</configId>
      <percentage>100.0</percentage>
    </distributions>
  </distributions>
</PercentageBean>
```

Update EnvModelPercentage

PUT <http://<host>:<port>/updates/percentageBean?applicationType={type}>

Headers:

Content-Type = application/json OR application/xml

Accept = application/json OR application/xml

applicationType param is not required, default value is stb

Response: 200 OK, 404 NOT FOUND, 409 CONFLICT, 400 BAD REQUEST

Request: <http://<host>:<port>/queries/percentageBean>

Restrictions:

Name should be unique and not blank, environment and model should be not empty, at least one firmware version should be in minCheck list if firmwareCheckRequired=true, percentage within [0, 100], distribution firmware version should be in minCheck list if firmwareCheckRequired=true, total distribution percentage is within [0, 100], last known good is not empty if total distribution percentage < 100, last known good should be selected in minCheck list.

JSON request


```
{
  "id": "4c2cea9e-c1be-4219-8f2a-70084f1cb648",
  "name": "Percent Test 2",
  "active": false,
  "firmwareCheckRequired": false,
  "rebootImmediately": false,
  "firmwareVersions": [],
  "environment": "envId",
  "model": "modelId",
  "distributions": [
    {
      "configId": "0c176c01-b4cf-4706-9e41-5118e30f68cb",
      "percentage": 100.0
    }
  ]
}
```

XML request

```
<PercentageBean>
  <id>4c2cea9e-c1be-4219-8f2a-70084f1cb648</id>
  <ttlMap/>
  <name>Percent Test 2</name>
  <active>false</active>
  <firmwareCheckRequired>false</firmwareCheckRequired>
  <rebootImmediately>false</rebootImmediately>
  <firmwareVersions/>
  <environment>envId</environment>
  <model>modelId</model>
  <distributions>
    <distributions>
      <configId>0c176c01-b4cf-4706-9e41-5118e30f68cb</configId>
      <percentage>100.0</percentage>
    </distributions>
  </distributions>
</PercentageBean>
```

Delete envModelPercentage

DELETE http://<host>:<port>/delete/percentageBean/id

Headers:

Content-Type = application/json OR application/xml

Accept = application/json OR application/xml

Response: 204 NO CONTENT OR 404 NOT FOUND

Request example: <http://<host>:<port>/delete/percentageBean/testId>

Time filter

Retrieve time filter list

GET <http://<host>:<port>/queries/filters/time?applicationType={type}>

Headers:

Accept = application/json OR application/xml

applicationType param is not required, default value is stb

Response: 200 OK, 400 BAD REQUEST

Request: <http://<host>:<port>/queries/filters/time.json> OR <http://<host>:<port>/queries/filters/time>

JSON response

```
[
  {
    "id": "cef2b340-2109-467e-a5d2-a70f0cddb432",
    "name": "abc",
    "ipWhitelist": {
      "id": "0e277b6a-7d95-4421-b390-415b03de04b4",
      "name": "60.70.80.90",
      "ipAddresses": [
        "60.70.80.90"
      ]
    },
    "envModelWhitelist": {
      "id": "0f8110a3-4b60-489b-854e-c69e25b18abc",
      "environmentId": "QA",
      "modelId": "SR150BNC"
    },
    "neverBlockRebootDecoupled": true,
    "neverBlockHttpDownload": true,
    "startTime": "01:00",
    "endTime": "02:00",
    "localTime": true
  }
]
```

XML response

```

<ArrayList>

  <item>

    <id>cef2b340-2109-467e-a5d2-a70f0cddb432</id>

    <name>abc</name>

    <ipWhitelist>

      <id>0e277b6a-7d95-4421-b390-415b03de04b4</id>

      <name>60.70.80.90</name>

      <ipAddresses>

        <ipAddresses>60.70.80.90</ipAddresses>

      </ipAddresses>

    </ipWhitelist>

    <envModelWhitelist>

      <id>0f8110a3-4b60-489b-854e-c69e25b18abc</id>

      <environmentId>QA</environmentId>

      <modelId>SR150BNC</modelId>

    </envModelWhitelist>

    <neverBlockRebootDecoupled>true</neverBlockRebootDecoupled>

    <neverBlockHttpDownload>true</neverBlockHttpDownload>

    <startTime>01:00</startTime>

    <endTime>02:00</endTime>

    <localTime>true</localTime>

  </item>

</ArrayList>

```

Retrieve time filter by name

GET <http://<host>:<port>/queries/filters/time/{name}?applicationType={type}>

Headers:

Accept = application/json OR application/xml

applicationType param is not required, default value is stb

Response: 200 OK, 400 BAD REQUEST

Request: <http://<host>:<port>/queries/filters/time/xcc.td.json> OR <http://<host>:<port>/queries/filters/time/xcc.td>

JSON response

```
{
  "id": "cef2b340-2109-467e-a5d2-a70f0cddb432",
  "name": "abc",
  "ipWhitelist": {
    "id": "0e277b6a-7d95-4421-b390-415b03de04b4",
    "name": "60.70.80.90",
    "ipAddresses": [
      "60.70.80.90"
    ]
  },
  "envModelWhitelist": {
    "id": "0f8110a3-4b60-489b-854e-c69e25b18abc",
    "environmentId": "QA",
    "modelId": "SR150BNC"
  },
  "neverBlockRebootDecoupled": true,
  "neverBlockHttpDownload": true,
  "startTime": "01:00",
  "endTime": "02:00",
  "localTime": true
}
```

XML response

```

<TimeFilterWrapper>

  <id>cef2b340-2109-467e-a5d2-a70f0cddb432</id>

  <name>abc</name>

  <ipWhitelist>

    <id>0e277b6a-7d95-4421-b390-415b03de04b4</id>

    <name>60.70.80.90</name>

    <ipAddresses>

      <ipAddresses>60.70.80.90</ipAddresses>

    </ipAddresses>

  </ipWhitelist>

  <envModelWhitelist>

    <id>0f8110a3-4b60-489b-854e-c69e25b18abc</id>

    <environmentId>QA</environmentId>

    <modelId>SR150BNC</modelId>

  </envModelWhitelist>

  <neverBlockRebootDecoupled>true</neverBlockRebootDecoupled>

  <neverBlockHttpDownload>true</neverBlockHttpDownload>

  <startTime>01:00</startTime>

  <endTime>02:00</endTime>

  <localTime>true</localTime>

</TimeFilterWrapper>

```

Create/update time filter

If time filter is missing it will be being created, otherwise updated. For update operation id field is not needed.

POST <http://<host>:<port>/updates/filters/time?applicationType={type}>

Headers:

Content-Type = application/json OR application/xml

Accept = application/json OR application/xml

applicationType param is not required, default value is stb

Response: 200 OK and saved object (see get request), 400 BAD REQUEST, 500 INTERNAL SERVER ERROR

Restrictions:

Name should be unique

Request: <http://<host>:<port>/updates/filters/time.json> OR <http://<host>:<port>/updates/filters/time>

JSON request

```
{
  "id": "cef2b340-2109-467e-a5d2-a70f0cddb432",
  "name": "abc",
  "ipWhitelist": {
    "id": "0e277b6a-7d95-4421-b390-415b03de04b4",
    "name": "60.70.80.90",
    "ipAddresses": [
      "60.70.80.90"
    ]
  },
  "envModelWhitelist": {
    "id": "0f8110a3-4b60-489b-854e-c69e25b18abc",
    "environmentId": "QA",
    "modelId": "SR150BNC"
  },
  "neverBlockRebootDecoupled": true,
  "neverBlockHttpDownload": true,
  "startTime": "01:00",
  "endTime": "02:00",
  "localTime": true
}
```

XML request

```
<TimeFilterWrapper>
  <id>cef2b340-2109-467e-a5d2-a70f0cddb432</id>
  <name>abc</name>
  <ipWhitelist>
    <id>0e277b6a-7d95-4421-b390-415b03de04b4</id>
    <name>60.70.80.90</name>
    <ipAddresses>
      <ipAddresses>60.70.80.90</ipAddresses>
    </ipAddresses>
  </ipWhitelist>
  <envModelWhitelist>
    <id>0f8110a3-4b60-489b-854e-c69e25b18abc</id>
    <environmentId>QA</environmentId>
    <modelId>SR150BNC</modelId>
  </envModelWhitelist>
  <neverBlockRebootDecoupled>true</neverBlockRebootDecoupled>
  <neverBlockHttpDownload>true</neverBlockHttpDownload>
  <startTime>01:00</startTime>
  <endTime>02:00</endTime>
  <localTime>true</localTime>
</TimeFilterWrapper>
```

Delete time filter by name

DELETE <http://<host>:<port>/delete/filters/time/{timeFilterName}?applicationType={type}>

Headers:

Accept = application/json OR application/xml

applicationType is not required, default value is stb

Response: 204 NO CONTENT and message: Time Filter successfully deleted OR Filter doesn't exist with name: <filterName>

Request example: <http://<host>:<port>/delete/filters/time/name>

Environment

Retrieve an list of environments

GET <http://<host>:<port>/queries/environments>

Headers:

Accept = application/json OR application/xml

Response: 200 OK

Request: <http://<host>:<port>/queries/environments> OR <http://<host>:<port>/queries/environments.json>

XML response

```
<ArrayList>
  <item>
    <id>DEV</id>
    <description>ff</description>
  </item>
  <item>
    <id>TEST</id>
    <description>do not delete</description>
  </item>
</ArrayList>
```

JSON response

```
[{"id":"DEV","description":"ff"}, {"id":"TEST","description":"do not delete"}]
```

Retrieve environment by id

GET <http://<host>:<port>/queries/environments/<environmentId>>

Headers:

Accept = application/json OR application/xml

Response: 200 OK; 400 BAD REQUEST

Request: <http://<host>:<port>/queries/environments/DEV> OR <http://<host>:<port>/queries/environments/DEV.json>

JSON response

```
{"id":"DEV","description":"ff"}
```

XML response

```
<Environment>
  <id>DEV</id>
  <description>ff</description>
</Environment>
```

Create an environment

POST <http://<host>:<port>/updates/environments>

Headers:

Content-Type: application/json OR application/xml

Accept = application/json OR application/xml

Response: 200 OK and saved object (see get request); 400 BAD REQUEST; 500 INTERNAL SERVER ERROR

Restrictions:

Environment name should be valid by pattern: `^[a-zA-Z0-9]+$`, name should be unique

Request: <http://<host>:<port>/updates/environments> OR <http://<host>:<port>/updates/environments.json>

XML request

```
<Environment>
  <id>testName</id>
  <description>some description</description>
</Environment>
```

JSON request

```
{"id":"testName","description":"some description"}
```

Delete environment by id

DELETE <http://<host>:<port>/delete/environments/<environmentId>>

Headers:

Accept = application/json OR application/xml

Response: 204 NO CONTENT and message: Environment doesn't exist OR Environment successfully deleted; 400 BAD REQUEST: Environment is used:
<usage place>

Restrictions:

Environment should be not used

Request: <http://<host>:<port>/delete/environments/DEV>

IP address group

Retrieve an IP address group list

GET <http://<host>:<port>/queries/ipAddressGroups>

Headers:

Accept = application/json OR application/xml

Response: 200 OK

Request: Request: <http://<host>:<port>/queries/ipAddressGroups> OR <http://<host>:<port>/queries/ipAddressGroups.json>

XML response

```
<ArrayList>
  <item>
    <id>2c184325-f9eb-4edc-85c3-5b6466fc3c5c</id>
    <name>test</name>
    <ipAddresses>
      <ipAddresses>192.11.11.11</ipAddresses>
    </ipAddresses>
  </item>
</ArrayList>
```

JSON response

```
[
  {
    "id": "2c184325-f9eb-4edc-85c3-5b6466fc3c5c",
    "name": "test",
    "ipAddresses": [
      "192.11.11.11"
    ]
  }
]
```

Retrieve an IP address group by name

GET <http://<host>:<port>/queries/ipAddressGroups/byName/<ipAddressGroupName>/>

Headers:

Accept = application/json OR application/xml

Response: 200 OK; 400 BAD REQUEST

Request: <http://<host>:<port>/queries/ipAddressGroups/byName/test/.json> OR <http://<host>:<port>/queries/ipAddressGroups/byName/test/>

JSON response

```
[{"id":"2c184325-f9eb-4edc-85c3-5b6466fc3c5c","name":"test","ipAddresses":["192.11.11.11"]}]
```

XML response

```
<ArrayList>
  <item>
    <id>2c184325-f9eb-4edc-85c3-5b6466fc3c5c</id>
    <name>test</name>
    <ipAddresses>
      <ipAddresses>192.11.11.11</ipAddresses>
    </ipAddresses>
  </item>
</ArrayList>
```

Retrieve an IP address group by IP

GET <http://<host>:<port>/queries/ipAddressGroups/byIp/<ipAddressGroupIp>/>

Headers:

Accept = application/json OR application/xml

Response: 200 OK; 400 BAD REQUEST

Request: <http://<host>:<port>/queries/ipAddressGroups/byIp/192.11.11.121/json> OR <http://<host>:<port>/queries/ipAddressGroups/byIp/192.11.11.121/>

JSON response

```
[{"id":"2c184325-f9eb-4edc-85c3-5b6466fc3c5c","name":"test","ipAddresses":["192.11.11.11"]}]
```

XML response

```
<ArrayList>
  <item>
    <id>2c184325-f9eb-4edc-85c3-5b6466fc3c5c</id>
    <name>test</name>
    <ipAddresses>
      <ipAddresses>192.11.11.11</ipAddresses>
    </ipAddresses>
  </item>
</ArrayList>
```

Create an IP address group

POST <http://<host>:<port>/updates/ipAddressGroups>

Headers:

Content-Type: application/json OR application/xml

Accept = application/json OR application/xml

Response: 200 OK and saved object (see get request); 400 BAD REQUEST; 500 INTERNAL SERVER ERROR

Restrictions:

Name should be not empty and unique

Request: <http://<host>:<port>/updates/ipAddressGroups.json> OR <http://<host>:<port>/updates/ipAddressGroups>

XML request

```
<IpAddressGroupExtended>
  <id>2c184325-f9eb-4edc-85c3-5b6466fc3c5c</id>
  <name>test</name>
  <ipAddresses>
    <ipAddresses>192.11.11.11</ipAddresses>
  </ipAddresses>
</IpAddressGroupExtended>
```

JSON request

```
{"id":"2c184325-f9eb-4edc-85c3-5b6466fc3c5c","name":"test","ipAddresses":["192.11.11.11"]}
```

Add data to IP Address Group(dev in progress)

POST http://<host>:<port>/updates/ipAddressGroups/<ipAddressGroup_name>/addData

Headers:

Content-Type = application/json OR application/xml

Accept = application/json OR application/xml

Response: 200 OK and ipAddressGroup object (see get request); 400 BAD REQUEST; 500 INTERNAL SERVER ERROR

Restrictions:

ipAddressGroup with current id should be exist, see restrictions for create operation

Request: http://<host>:<port>/updates/ipAddressGroups/ipAddressGroup_name/addData.json OR http://<host>:<port>/updates/ipAddressGroups/ipAddressGroup_name/addData

JSON request

```
{
  "list": [
    "1.1.1.1"
  ]
}
```

XML request

```
<StringListWrapper>
  <list>
    <list>1.1.1.1</list>
  </list>
</StringListWrapper>
```

Delete data from IP Address Group(dev in progress)

POST http://<host>:<port>/updates/ipAddressGroups/<ipAddressGroup_name>/removeData

Headers:

Content-Type = application/json OR application/xml

Accept = application/json OR application/xml

Response: 204 NO CONTENT and ipAddressGroup object (see get request); 400 BAD REQUEST

Restrictions:

List contains IPs, which should be present in current Namespaced list, namespaced list should contain at least one IP address, and restrictions for create operation

Request: http://<host>:<port>/updates/ipAddressGroups/ipAddressGroup_name/removeData.json OR http://<host>:<port>/updates/ipAddressGroups/ipAddressGroup_name/removeData

JSON request

```
{
  "list": [
    "1.1.1.1"
  ]
}
```

XML request

```
<StringListWrapper>
  <list>
    <list>1.1.1.1</list>
  </list>
</StringListWrapper>
```

Delete an IP address group by id

DELETE <http://<host>:<port>/delete/ipAddressGroups/<ipAddressGroupId>>

Headers:

Accept = application/json OR application/xml

Response: 204 NO CONTENT and message: IpAddressGroup doesn't exist OR IpAddressGroup successfully deleted; 400 BAD REQUEST: IpAddressGroup is used: <usage place>

Restrictions:

IP address group should be not used

Request: <http://<host>:<port>/delete/ipAddressGroups/607b315b-c744-4c43-87fa-51b17aa92b09>

Mac rule

Retrieve a mac rule list (legacy)

GET <http://<host>:<port>/queries/rules/macs?applicationType={type}>

Headers:

Accept = application/json OR application/xml

applicationType is not required, default value is stb

With no version parameter or version < 2. Legacy query.

Response: 200 OK, 400 BAD REQUEST

Request: <http://<host>:<port>/queries/rules/macs.json> OR <http://<host>:<port>/queries/rules/macs>

JSON response

```
[
  {
    "name": "test",
    "macListRef": "macs",
    "targetedModelIds": [
      "YETST"
    ],
    "firmwareConfig": {
      "id": "b65962b5-1481-4eed-a010-2abfa8c3bbfd",
      "description": "_-",
      "supportedModelIds": [
        "YETST"
      ],
      "firmwareFilename": "_-",
      "firmwareVersion": "_-"
    }
  }
]
```

XML response

```
<ArrayList>
  <item>
    <name>test</name>
    <macListRef>macs</macListRef>
    <targetedModelIds>
      <targetedModelIds>YETST</targetedModelIds>
    </targetedModelIds>
    <firmwareConfig>
      <id>b65962b5-1481-4eed-a010-2abfa8c3bbfd</id>
      <description>_-</description>
      <supportedModelIds>
        <supportedModelIds>YETST</supportedModelIds>
      </supportedModelIds>
      <firmwareFilename>_-</firmwareFilename>
      <firmwareVersion>_-</firmwareVersion>
    </firmwareConfig>
  </item>
</ArrayList>
```

Retrieve a mac rule list (v2)

GET <http://<host>:<port>/queries/rules/macs?version=2&applicationType={type}>

Headers:

Accept = application/json OR application/xml

applicationType is not required, default value is stb

Version parameter could any number >= 2.

Response: 200 OK, 400 BAD REQUEST

Request: <http://<host>:<port>/queries/rules/mac.json?version=2> OR <http://<host>:<port>/queries/rules/macs?version=2>

JSON response

```
[
  {
    "id": "18542fe1-94f2-487c-8f41-9f20965648f5",
    "name": "test",
    "macListRef": "macs",
    "targetedModelIds": [
      "YETST"
    ],
    "firmwareConfig": {
      "id": "b65962b5-1481-4eed-a010-2abfa8c3bbfd",
      "description": "_-",
      "supportedModelIds": [
        "YETST"
      ],
      "firmwareFilename": "_-",
      "firmwareVersion": "_-"
    },
    "macList": [
      "AA:AA:AA:AA:AA:AA"
    ]
  }
]
```

XML response

```
<ArrayList>
  <item>
    <id>18542fe1-94f2-487c-8f41-9f20965648f5</id>
    <name>test</name>
    <macListRef>macs</macListRef>
    <targetedModelIds>
      <targetedModelIds>YETST</targetedModelIds>
    </targetedModelIds>
    <firmwareConfig>
      <id>b65962b5-1481-4eed-a010-2abfa8c3bbfd</id>
      <description>_</description>
      <supportedModelIds>
        <supportedModelIds>YETST</supportedModelIds>
      </supportedModelIds>
      <firmwareFilename>_</firmwareFilename>
      <firmwareVersion>_</firmwareVersion>
    </firmwareConfig>
    <macList>
      <macList>AA:AA:AA:AA:AA:AA</macList>
    </macList>
  </item>
</ArrayList>
```

Retrieve mac rule by name (legacy)

GET <http://<host>:<port>/queries/rules/macs/{macRuleName}?applicationType={type}>

Headers:

Accept = application/json OR application/xml

applicationType is not required, default value is stb

With no version parameter or version < 2. Legacy query.

Response: 200 OK;

Request: <http://<host>:<port>/queries/rules/macs/test.json> OR <http://<host>:<port>/queries/rules/macs/test>

JSON response

```
[
  {
    "name": "test",
    "macListRef": "macs",
    "targetedModelIds": [
      "YETST"
    ],
    "firmwareConfig": {
      "id": "b65962b5-1481-4eed-a010-2abfa8c3bbfd",
      "description": "_-",
      "supportedModelIds": [
        "YETST"
      ],
      "firmwareFilename": "_-",
      "firmwareVersion": "_-"
    }
  }
]
```

XML response

```
<MacRuleBeanWrapper>
  <name>test</name>
  <macListRef>macs</macListRef>
  <targetedModelIds>
    <targetedModelIds>YETST</targetedModelIds>
  </targetedModelIds>
  <firmwareConfig>
    <id>b65962b5-1481-4eed-a010-2abfa8c3bbfd</id>
    <description>_-</description>
    <supportedModelIds>
      <supportedModelIds>YETST</supportedModelIds>
    </supportedModelIds>
    <firmwareFilename>_-</firmwareFilename>
    <firmwareVersion>_-</firmwareVersion>
  </firmwareConfig>
</MacRuleBeanWrapper>
```

Retrieve mac rule by name (v2)

GET <http://<host>:<port>/queries/rules/macs/<macRuleName>?version=2&applicationType={type}>

Headers:

Accept = application/json OR application/xml

applicationType is not required, default value is stb

Version parameter could any number >= 2.

Response: 200 OK;

Request: <http://<host>:<port>/queries/rules/macs/test.json?version=2> OR <http://<host>:<port>/queries/rules/macs/test?version=2>

JSON response

```
[
  {
    "id": "18542fe1-94f2-487c-8f41-9f20965648f5",
    "name": "test",
    "macListRef": "macs",
    "targetedModelIds": [
      "YETST"
    ],
    "firmwareConfig": {
      "id": "b65962b5-1481-4eed-a010-2abfa8c3bbfd",
      "description": "_-",
      "supportedModelIds": [
        "YETST"
      ],
      "firmwareFilename": "_-",
      "firmwareVersion": "_-"
    },
    "macList": [
      "AA:AA:AA:AA:AA:AA"
    ]
  }
]
```

XML response

```
<MacRuleBeanWrapper>
  <id>18542fe1-94f2-487c-8f41-9f20965648f5</id>
  <name>test</name>
  <macListRef>macs</macListRef>
  <targetedModelIds>
    <targetedModelIds>YETST</targetedModelIds>
  </targetedModelIds>
  <firmwareConfig>
    <id>b65962b5-1481-4eed-a010-2abfa8c3bbfd</id>
    <description>_</description>
    <supportedModelIds>
      <supportedModelIds>YETST</supportedModelIds>
    </supportedModelIds>
    <firmwareFilename>_</firmwareFilename>
    <firmwareVersion>_</firmwareVersion>
  </firmwareConfig>
  <macList>
    <macList>AA:AA:AA:AA:AA:AA</macList>
  </macList>
</MacRuleBeanWrapper>
```

Retrieve mac rule by mac address (legacy)

GET <http://<host>:<port>/queries/rules/macs/address/{macAddress}?applicationType={type}>

Headers:

Accept = application/json OR application/xml

applicationType is not required, default value is stb

With no version parameter or version < 2. Legacy query.

Response: 200 OK, 400 BAD REQUEST

Request: <http://<host>:<port>/queries/rules/macs/address/AA:AA:AA:AA:AA:AA.json> OR <http://<host>:<port>/queries/rules/macs/address/AA:AA:AA:AA:AA:AA>

JSON response

```
[
  {
    "name": "test",
    "macListRef": "macs",
    "targetedModelIds": [
      "YETST"
    ],
    "firmwareConfig": {
      "id": "b65962b5-1481-4eed-a010-2abfa8c3bbfd",
      "description": "_-",
      "supportedModelIds": [
        "YETST"
      ],
      "firmwareFilename": "_-",
      "firmwareVersion": "_-"
    }
  }
]
```

XML response

```
<ArrayList>
  <item>
    <name>test</name>
    <macListRef>macs</macListRef>
    <targetedModelIds>
      <targetedModelIds>YETST</targetedModelIds>
    </targetedModelIds>
    <firmwareConfig>
      <id>b65962b5-1481-4eed-a010-2abfa8c3bbfd</id>
      <description>_-</description>
      <supportedModelIds>
        <supportedModelIds>YETST</supportedModelIds>
      </supportedModelIds>
      <firmwareFilename>_-</firmwareFilename>
      <firmwareVersion>_-</firmwareVersion>
    </firmwareConfig>
  </item>
</ArrayList>
```

Retrieve mac rule by mac address (v2)

GET <http://<host>:<port>/queries/rules/macs/address/<macAddress>?version=2&applicationType={type}>

Headers:

Accept = application/json OR application/xml

applicationType is not required, default value is stb

Version parameter could any number >= 2.

Response: 200 OK, 400 BAD REQUEST

Request: <http://<host>:<port>/queries/rules/macs/address/AA:AA:AA:AA:AA:AA.json?version=2> OR <http://<host>:<port>/queries/rules/macs/address/AA:AA:AA:AA:AA:AA?version=2>

JSON response

```
[
  {
    "id": "18542fe1-94f2-487c-8f41-9f20965648f5",
    "name": "test",
    "macListRef": "macs",
    "targetedModelIds": [
      "YETST"
    ],
    "firmwareConfig": {
      "id": "b65962b5-1481-4eed-a010-2abfa8c3bbfd",
      "description": "_-",
      "supportedModelIds": [
        "YETST"
      ],
      "firmwareFilename": "_-",
      "firmwareVersion": "_-"
    },
    "macList": [
      "AA:AA:AA:AA:AA:AA"
    ]
  }
]
```

XML response

```

<ArrayList>
  <item>
    <id>18542fe1-94f2-487c-8f41-9f20965648f5</id>
    <name>test</name>
    <macListRef>macs</macListRef>
    <targetedModelIds>
      <targetedModelIds>YETST</targetedModelIds>
    </targetedModelIds>
    <firmwareConfig>
      <id>b65962b5-1481-4eed-a010-2abfa8c3bbfd</id>
      <description>_</description>
      <supportedModelIds>
        <supportedModelIds>YETST</supportedModelIds>
      </supportedModelIds>
      <firmwareFilename>_</firmwareFilename>
      <firmwareVersion>_</firmwareVersion>
    </firmwareConfig>
    <macList>
      <macList>AA:AA:AA:AA:AA:AA</macList>
    </macList>
  </item>
</ArrayList>

```

Create/update mac rule

If mac rule is missing it will be being created, otherwise updated. For update operation id field is not needed.

POST <http://<host>:<port>/updates/rules/macs?applicationType={type}>

Headers:

Content-Type = application/json OR application/xml

Accept = application/json OR application/xml

applicationType is not required, default value is stb

Response: 200 OK; 201 CREATED; 400 BAD REQUEST; 500 INTERNAL SERVER ERROR

Restrictions:

Name, mac address list, model list, mac list, firmware configuration should be not empty, mac address list is never used in another rule, model list contain only existed model, firmware config should support given models

Request: <http://<host>:<port>/updates/rules/macs.json> OR <http://<host>:<port>/updates/rules/macs>

JSON request

```
{
  "id": "18542fe1-94f2-487c-8f41-9f20965648f5",
  "name": "test",
  "macListRef": "macs",
  "targetedModelIds": [
    "YETST"
  ],
  "firmwareConfig": {
    "id": "b65962b5-1481-4eed-a010-2abfa8c3bbfd",
    "description": "_-",
    "supportedModelIds": [
      "YETST"
    ],
    "firmwareFilename": "_-",
    "firmwareVersion": "_-"
  },
  "macList": [
    "AA:AA:AA:AA:AA:AA"
  ]
}
```

XML request

```
<MacRuleBeanWrapper>
  <id>18542fe1-94f2-487c-8f41-9f20965648f5</id>
  <name>test</name>
  <macListRef>macs</macListRef>
  <targetedModelIds>
    <targetedModelIds>YETST</targetedModelIds>
  </targetedModelIds>
  <firmwareConfig>
    <id>b65962b5-1481-4eed-a010-2abfa8c3bbfd</id>
    <description>_-</description>
    <supportedModelIds>
      <supportedModelIds>YETST</supportedModelIds>
    </supportedModelIds>
    <firmwareFilename>_-</firmwareFilename>
    <firmwareVersion>_-</firmwareVersion>
  </firmwareConfig>
  <macList>
    <macList>AA:AA:AA:AA:AA:AA</macList>
  </macList>
</MacRuleBeanWrapper>
```

Delete mac rule by name

DELETE <http://<host>:<port>/delete/rules/mac/{macRuleName}?applicationType={type}>

Headers:

Accept = application/json OR application/xml

applicationType is not required, default value is stb

Response: 204 NO CONTENT and message: MacRule does'n exist OR MacRule deleted successfully, 400 BAD REQUEST

Request example: <http://<host>:<port>/delete/rules/macs/macRuleName>

Model

Retrieve a model list

GET <http://<host>:<port>/queries/models>

Headers:

Accept = application/json OR application/xml

Response: 200 OK

Request: <http://<host>:<port>/queries/models.json> OR <http://<host>:<port>/queries/models>

JSON response

```
[
  {
    "id": "YETST",
    "description": ""
  },
  {
    "id": "PX013ANC",
    "description": "Pace XG1v3 - Cisco Cable Card"
  }
]
```

XML response

```
<ArrayList>
  <item>
    <id>YETST</id>
    <description></description>
  </item>
  <item>
    <id>PX013ANC</id>
    <description>Pace XG1v3 - Cisco Cable Card</description>
  </item>
</ArrayList>
```

Retrieve model by id

GET <http://<host>:<port>/queries/models/<modelId>>

Headers:

Accept = application/json OR application/xml

Response: 200 OK; 204 NO CONTENT

Request: <http://<host>:<port>/queries/models/YETST.json> OR <http://<host>:<port>/queries/models/YETST>

JSON response

```
{
  "id": "YETST",
  "description": ""
}
```

XML response

```
<Model>
  <id>YETST</id>
  <description></description>
</Model>
```

Create model

POST <http://<host>:<port>/updates/models>

Headers:

Content-Type = application/json OR application/xml

Accept = application/json OR application/xml

Response: 201 CREATED; 400 BAD REQUEST; 500 INTERNAL SERVER ERROR

Restrictions:

Model name should be unique and valid by pattern: `^[a-zA-Z0-9]+$`

Request: <http://<host>:<port>/updates/models.json> OR <http://<host>:<port>/updates/models>

JSON request

```
{
  "id": "test",
  "description": ""
}
```

XML request

```
<Model>
  <id>YETST</id>
  <description></description>
</Model>
```

Update Model Description

PUT <http://<host>:<port>/updates/models>

Headers: Content-Type = application/json OR application/xml Accept = application/json OR application/xml

Response: 200 OK; 400 BAD REQUEST; 404 NOT FOUND, 500 INTERNAL SERVER ERROR

Request: <http://localhost:9091/updates/models.json> OR <http://localhost:9091/updates/models>

JSON Request

```
{
  "id": "test",
  "description": "description update"
}
```

XML Request

```
<Model><id>test</id><description>description update</description></Model>
```

Delete model by id

DELETE <http://<host>:<port>/delete/models/<modelId>>

Headers:

Accept = application/json OR application/xml

Response: 204 NO CONTENT and message: Model doesn't exist OR Model deleted successfully;

Restrictions:

Model should be not used in another places

Request: <http://<host>:<port>/delete/models/YETST>

NamespacedList

Retrieve all NS lists

GET <http://<host>:<port>/queries/nsLists>

Headers:

Accept = application/json OR application/xml

Response: 200 OK

Request: <http://<host>:<port>/queries/nsLists.json> OR <http://<host>:<port>/queries/nsLists>

JSON response

```
[
  {
    "id": "macs",
    "data": [
      "AA:AA:AA:AA:AA:AA"
    ]
  }
]
```

XML response

```
<ArrayList>
<item>
  <id>macs</id>
  <data>
    <data>AA:AA:AA:AA:AA:AA</data>
  </data>
</item>
</ArrayList>
```

Retrieve an NS list by id:

GET <http://<host>:<port>/queries/nsLists/byId/<nsListId>>

Headers:

Accept = application/json OR application/xml

Response: 200 OK

Request: <http://<host>:<port>/queries/nsLists/byId/macs.json> OR <http://<host>:<port>/queries/nsLists/byId/macs>

JSON response

```
{
  "id": "macs",
  "data": [
    "AA:AA:AA:AA:AA:AA"
  ]
}
```

XML response

```
<NamespacedList>
  <id>macs</id>
  <data>
    <data>AA:AA:AA:AA:AA:AA</data>
  </data>
</NamespacedList>
```

Retrieve NS list by mac part

GET <http://<host>:<port>/queries/nsLists/byMacPart/<macAddressPart>>

Headers:

Accept = application/json OR application/xml

Response: 200 OK

Request: <http://<host>:<port>/queries/nsLists/byMacPart/AA:AA.json> OR <http://<host>:<port>/queries/nsLists/byMacPart/AA:AA>

JSON response

```
{
  "id": "macs",
  "data": [
    "AA:AA:AA:AA:AA:AA"
  ]
}
```

XML response

```
<NamespacedList>
  <id>macs</id>
  <data>
    <data>AA:AA:AA:AA:AA:AA</data>
  </data></NamespacedList>
```

Create a NS list

POST <http://<host>:<port>/updates/nsLists>

Headers:

Content-Type = application/json OR application/xml

Accept = application/json OR application/xml

Response: 201 CREATED; 400 BAD REQUEST; 500 INTERNAL SERVER ERROR

Restrictions:

Name should be valid by pattern: `^[a-zA-Z0-9]+$`, list data should be not empty and contain valid mac addresses, mac address should be used only in one NS list

Request: <http://<host>:<port>/updates/nsLists.json> OR <http://<host>:<port>/updates/nsLists>

JSON request

```
{
  "id": "macs",
  "data": [
    "AA:AA:AA:AA:AA:AA"
  ]
}
```

XML request

```
<NamespacedList>
  <id>macs</id>
  <data>
    <data>AA:AA:AA:AA:AA:AA</data>
  </data>
</NamespacedList>
```

Add data to NS list

POST <http://<host>:<port>/updates/nsLists/<nsListId>/addData>

POST <http://<host>:<port>/updates/nslist/<nsListId>/addData> - legacy endpoint

Headers:

Content-Type = application/json OR application/xml

Accept = application/json OR application/xml

Response: 200 OK and NS list object (see get request); 400 BAD REQUEST; 500 INTERNAL SERVER ERROR

Restrictions:

NS list with current id should be exist, see restrictions for create operation

Request: <http://<host>:<port>/updates/nsLists/listName/addData.json> OR <http://<host>:<port>/updates/nsLists/listName/addData>

JSON request

```
{
  "list": [
    "11:22:33:44:55:66"
  ]
}
```

XML request

```
<StringListWrapper>
  <list>
    <list>11:22:33:44:55:66</list>
  </list>
</StringListWrapper>
```

Delete data from NS list

DELETE <http://<host>:<port>/updates/nsLists/<nsListId>/removeData>

DELETE <http://<host>:<port>/updates/nslist/<nsListId>/removeData> - legacy endpoint

Headers:

Content-Type = application/json OR application/xml

Accept = application/json OR application/xml

Response: 204 NO CONTENT and NS list object (see get request); 400 BAD REQUEST

Restrictions:

List contains MACs, which should be present in current Namespaced list, namespaced list should contain at least one MAC address, and restrictions for create operation

Request: <http://<host>:<port>/updates/nsLists/listName/removeData.json> OR <http://<host>:<port>/updates/nsLists/listName/removeData>

JSON request

```
{
  "list": [
    "11:22:33:44:55:66"
  ]
}
```

XML request

```
<StringListWrapper>
  <list>
    <list>11:22:33:44:55:66</list>
  </list>
</StringListWrapper>
```

Delete an NS list by id

DELETE <http://<host>:<port>/delete/nsLists/<nsListId>>

Headers:

Accept = application/json OR application/xml

Response: 200 OK and message: NamespacedList doesn't exist OR NamespacedList successfully deleted;

Restrictions:

NS list should be not used in another places

Request: <http://<host>:<port>/delete/nsLists/listName.json> OR <http://<host>:<port>/delete/nsLists/listName>

RebootImmediately filter

Retrieve an RI filter list

GET <http://<host>:<port>/queries/filters/ri?applicationType={type}>

Headers:

Accept = application/json OR application/xml

applicationType is not required, default value is stb

Response: 200 OK, 400 BAD REQUEST

Request: <http://<host>:<port>/queries/filters/ri.json> OR <http://<host>:<port>/queries/filters/ri>

JSON response

```
[
  {
    "macAddresses": "[AA:AA:AA:AA:AA:AA]",
    "environments": [
      "TEST"
    ],
    "models": [
      "YETST"
    ],
    "id": "7f6d54d8-69e1-44c3-bb28-e1ed2f5b016f",
    "name": "u"
  }
]
```

XML response

```
<ArrayList>
  <item>
    <macAddresses>AA:AA:AA:AA:AA:AA</macAddresses>
    <environments>
      <environments>TEST</environments>
    </environments>
    <models>
      <models>YETST</models>
    </models>
    <id>7f6d54d8-69e1-44c3-bb28-e1ed2f5b016f</id>
    <name>u</name>
  </item>
</ArrayList>
```

Retrieve and RI filter by rule name

GET <http://<host>:<port>/queries/filters/ri/{ruleName}?applicationType={type}>

Headers:

Accept = application/json OR application/xml

applicationType is not required, default value is stb

Response: 200 OK, 400 BAD REQUEST

Request: <http://<host>:<port>/queries/filters/ri/testName.json> OR <http://<host>:<port>/queries/filters/ri/testName>

JSON response

```
{
  "macAddresses": "AA:AA:AA:AA:AA:AA",
  "environments": [
    "TEST"
  ],
  "models": [
    "YETST"
  ],
  "id": "7f6d54d8-69e1-44c3-bb28-e1ed2f5b016f",
  "name": "testName"
}
```

XML response

```
<RebootImmediatelyFilter>
  <macAddresses>AA:AA:AA:AA:AA:AA</macAddresses>
  <environments>
    <environments>TEST</environments>
  </environments>
  <models>
    <models>YETST</models>
  </models>
  <id>7f6d54d8-69e1-44c3-bb28-e1ed2f5b016f</id>
  <name>u</name>
</RebootImmediatelyFilter>
```

Create/update an RI filter

If RI filter is missing it will be being created, otherwise updated. For update operation id field is not needed.

POST <http://<host>:<port>/updates/filters/ri?applicationType={type}>

Headers:

Accept = application/json OR application/xml

applicationType is not required, default value is stb

Response: 200 OK; 201 CREATED; 400 BAD REQUEST; 500 INTERNAL SERVER ERROR

Restrictions:

Name should be not empty, at least one of filter criteria should be specified, mac addresses should be valid

Request: <http://<host>:<port>/updates/filters/ri.json> OR <http://<host>:<port>/updates/filters/ri>

JSON request

```
{
  "macAddresses": "AA:AA:AA:AA:AA:AA",
  "environments": [
    "TEST"
  ],
  "models": [
    "YETST"
  ],
  "id": "7f6d54d8-69e1-44c3-bb28-e1ed2f5b016f",
  "name": "testName"
}
```

XML request

```
<RebootImmediatelyFilter>
  <macAddresses>AA:AA:AA:AA:AA:AA</macAddresses>
  <environments>
    <environments>TEST</environments>
  </environments>
  <models>
    <models>YETST</models>
  </models>
  <id>7f6d54d8-69e1-44c3-bb28-e1ed2f5b016f</id>
  <name>u</name>
</RebootImmediatelyFilter>
```

Delete RI filter by name

DELETE <http://<host>:<port>/delete/filters/ri/{riFilterName}?applicationType={type}>

Headers:

Accept = application/json OR application/xml

applicationType is not required, default value is stb

Response: 204 NO CONTENT and message: Filter doesn't exist OR Successfully deleted, 400 BAD REQUEST

Request: <http://<host>:<port>/delete/filters/ri/testName>

firmwareCheckRequired

```
{"firmwareVersions":["PR150BN_2.1p5s2_VBN_HYBse","PXD01ANI_2.1p10s1_PROD_sdy"]}
```

FirmwareRuleTemplate

Retrieve filtered templates

GET http://<host>:<port>/firmwareruletemplate/filtered?name=MAC_RULE&key=someKey

Headers: Accept = application/json, Content-Type = application/json

Request params:

1. Without params:
Retrieve all firmware rule templates.
2. 'name':
To filter templates by name.
3. 'key':
To filter by rule key
4. 'value':
to filter by rule value

Parameters could be combined:

'?name=someName&value=testValue'

Import firmware rule templates

POST http://<host>:<port>/firmwareruletemplate/importAll

Headers: Accept = application/json

Content-Type = application/json

Request body is list of firmware rule templates:

```
[
  {
    "id": "a1",
    "rule": {
      "negated": false,
      "condition": {
        "freeArg": {
          "type": "STRING",
          "name": "ipAddress"
        },
        "operation": "IS",
        "fixedArg": {
          "bean": {
            "value": {
              "java.lang.String": ""
            }
          }
        }
      },
      "compoundParts": []
    },
    "applicableAction": {
      "type": ".RuleAction",
      "ttlMap": {},
      "actionType": "RULE_TEMPLATE",
      "active": true,
      "firmwareCheckRequired": false,
      "rebootImmediately": false
    },
    "priority": 15,
    "requiredFields": [],
    "bypassFilters": [],
    "editable": true
  }
]
```

Return list of firmware rule templates' ids which were imported or not:

Response body:

```
{
  "NOT_IMPORTED": [],
  "IMPORTED": []
}
```

Response codes: 200, 400, 404, 409

FirmwareRule

Retrieve all firmware rules

GET http://<host>:<port>/firmwarerule Headers: Accept = application/json

Response body

```
[
  {
    "id": "7bc69543-c33a-42c5-97dc-7c21ecf06cdb",
    "name": "testName",
    "rule": {
      "negated": false,
      "condition": {
        "freeArg": {
          "type": "STRING",
          "name": "eStbMac"
        },
        "operation": "IN_LIST",
        "fixedArg": {
          "bean": {
            "value": {
              "java.lang.String": "mac list"
            }
          }
        }
      }
    },
    "applicableAction": {
      "type": ".DefinePropertiesAction",
      "ttlMap": {},
      "actionType": "DEFINE_PROPERTIES",
      "properties": {
        "rebootImmediately": "true"
      },
      "byPassFilters": [],
      "activationFirmwareVersions": {}
    },
    "type": "RI_MACLIST",
    "active": true,
    "applicationType": "stb"
  }
]
```

Retrieve filtered firmware rules

GET http://<host>:<port>/firmwarerule/filtered?templatId=TEST_ID&key=firmwareVersion

Headers: Accept = application/json, Content-Type = application/json

Request params:

'applicationType' param is required.

1. 'name':
'name=someName'
To filter templates by name.
2. 'key':
'key=someKey'
To filter by rule key
3. 'value':
'value=someValue'
to filter by rule value
4. 'firmwareVersion':
'firmwareVersion=testFirmwareVersion'
to filter by firmware version
5. 'templatId':
'templatId=testTemplate'
to filter by template
6. 'applicationType':
'applicationType=stb'
to filter by application type

Parameters could be combined:

'?applicationType=stb&value=someValue'

Import firmware rule

POST http://<host>:<port>/firmwarerule/importAll

Headers: Accept = application/json Content-Type = application/json

Request body is list of firmware rules:

```
[
  {
    "id": "7bc69543-c33a-42c5-97dc-7c21ecf06cdb",
    "name": "testName",
    "rule": {
      "negated": false,
      "condition": {
        "freeArg": {
          "type": "STRING",
          "name": "eStbMac"
        },
        "operation": "IN_LIST",
        "fixedArg": {
          "bean": {
            "value": {
              "java.lang.String": "mac list"
            }
          }
        }
      }
    },
    "applicableAction": {
      "type": ".DefinePropertiesAction",
      "ttlMap": {},
      "actionType": "DEFINE_PROPERTIES",
      "properties": {
        "rebootImmediately": "true"
      },
      "bypassFilters": [],
      "activationFirmwareVersions": {}
    },
    "type": "RI_MACLIST",
    "active": true,
    "applicationType": "stb"
  }
]
```

Return list of imported and not imported firmware rules' names:

Response body:

```
{
  "NOT_IMPORTED": [],
  "IMPORTED": ["testName"]
}
```

Response codes: 200, 400, 404, 409

Feature

Retrieve all features

GET http://<host>:<port>/feature

Headers: Accept = application/json

Response body

```
[
  {
    "id": "27a9d55e-0f0c-4809-b171-8357613f0efa",
    "name": "name",
    "featureName": "featureName",
    "effectiveImmediate": false,
    "enable": false,
    "whitelisted": false,
    "configData": {
      "key1": "value1"
    },
    "whitelistProperty": {},
    "applicationType": "stb"
  }
]
```

Retrieve filtered features

GET http://<host>:<port>/feature/filtered?

Headers: Accept = application/json, Content-Type = application/json

Request params:

'APPLICATION_TYPE' param is required.

1. NAME:
'NAME=someName'
To filter features by name
2. FEATURE_INSTANCE
'FEATURE_INSTANCE=someInstance'
To filter features by instance
3. FREE_ARG:
'FREE_ARG=someKey'
To filter by property key
4. FIXED_ARG:
'FIXED_ARG=someValue'
to filter by property value
5. APPLICATION_TYPE:
'APPLICATION_TYPE=stb'
to filter by application type

Parameters could be combined:

'?APPLICATION_TYPE=stb&FIXED_ARG=someValue'

Import feature

POST http://<host>:<port>/feature/importAll

Headers: Accept = application/json Content-Type = application/json

Request body is list of features:

```
[
  {
    "id": "27a9d55e-0f0c-4809-b171-8357613f0efa",
    "name": "name",
    "featureInstance": "featureInstance",
    "effectiveImmediate": false,
    "enable": false,
    "whitelisted": false,
    "configData": {
      "key1": "value1"
    },
    "whitelistProperty": {},
    "applicationType": "stb"
  }
]
```

Return list of imported and not imported feature ID's:

Response body:

```
{
  "NOT_IMPORTED": [],
  "IMPORTED": ["featureID"]
}
```

Response codes: 200, 400, 409

Create feature

POST <http://<host>:<port>/feature>

Headers: Accept = application/json Content-Type = application/json

Request body is feature json object

```
{
  "id": "27a9d55e-0f0c-4809-b171-8357613f0efa",
  "name": "name",
  "featureInstance": "featureInstance",
  "effectiveImmediate": false,
  "enable": false,
  "whitelisted": false,
  "configData": {
    "key1": "value1"
  },
  "whitelistProperty": {},
  "applicationType": "stb"
}
```

Response codes: 201, 400, 409.

Response body: created feature.

Update feature

PUT <http://<host>:<port>/feature>

Headers: Accept = application/json Content-Type = application/json

Request body is feature json object

```
{
  "id": "27a9d55e-0f0c-4809-b171-8357613f0efa",
  "name": "name",
  "featureInstance": "featureInstance",
  "effectiveImmediate": false,
  "enable": false,
  "whitelisted": false,
  "configData": {
    "key1": "value1"
  },
  "whitelistProperty": {},
  "applicationType": "stb"
}
```

Response codes: 201, 400, 404, 409.

Response body: created feature.

Delete feature

DELETE http://<host>:<port>/feature/{id}

Response codes: 204, 404, 409.

Feature Rule

Retrieve all feature rules

GET http://<host>:<port>/featurerule

Headers: Accept = application/json

Response body

```
[
  {
    "id": "963b2b7c-d198-4c23-820a-b56ac809c333",
    "name": "instance_test",
    "rule": {
      "negated": false,
      "condition": {
        "freeArg": {
          "type": "STRING",
          "name": "testKey"
        },
        "operation": "IS",
        "fixedArg": {
          "bean": {
            "value": {
              "java.lang.String": "testValue"
            }
          }
        }
      },
      "compoundParts": []
    },
    "priority": 1,
    "featureIds": [
      "27a9d55e-0f0c-4809-b171-8357613f0efa"
    ],
    "applicationType": "stb"
  }
]
```

Response codes: 200.

Retrieve filtered feature rules

GET http://<host>:<port>/featurerule/filtered?

Headers: Accept = application/json, Content-Type = application/json

Request params:

'APPLICATION_TYPE' param is required.

1. NAME:
'NAME=ruleName'
to filter feature rule by name
2. FREE_ARG:
'FREE_ARG=someKey'
to filter by feature rule key
3. FIXED_ARG:
'FIXED_ARG=someValue'
to filter by feature rule value
4. FEATURE:
'FEATURE=featureInstance'
to filter by feature instance.
5. APPLICATION_TYPE:
'APPLICATION_TYPE=stb'
to filter by application type

Parameters could be combined:

'?APPLICATION_TYPE=stb&FIXED_ARG=someValue'

Import feature rule

If feature rule with provided id does not exist it is imported otherwise updated.

POST http://<host>:<port>/featurerule/importAll

Headers: Accept = application/json Content-Type = application/json

Request body is list of feature rules:


```
[
  {
    "id": "963b2b7c-d198-4c23-820a-b56ac809c333",
    "name": "instance_test",
    "rule": {
      "negated": false,
      "condition": {
        "freeArg": {
          "type": "STRING",
          "name": "testKey"
        },
        "operation": "IS",
        "fixedArg": {
          "bean": {
            "value": {
              "java.lang.String": "testValue"
            }
          }
        }
      },
      "compoundParts": []
    },
    "priority": 1,
    "featureIds": [
      "27a9d55e-0f0c-4809-b171-8357613f0efa"
    ],
    "applicationType": "stb"
  }
]
```

Return list of imported and not imported feature rule IDs:

Response body:

```
{
  "NOT_IMPORTED": [],
  "IMPORTED": ["featureRuleId"]
}
```

Response codes: 200, 400, 404, 409

Create feature rule

POST http://<host>:<port>/featurerule

Headers: Accept = application/json Content-Type = application/json

Request body is feature rule json object

```
{
  "id": "963b2b7c-d198-4c23-820a-b56ac809c333",
  "name": "instance_test",
  "rule": {
    "negated": false,
    "condition": {
      "freeArg": {
        "type": "STRING",
        "name": "testKey"
      },
      "operation": "IS",
      "fixedArg": {
        "bean": {
          "value": {
            "java.lang.String": "testValue"
          }
        }
      }
    },
    "compoundParts": []
  },
  "priority": 1,
  "featureIds": [
    "27a9d55e-0f0c-4809-b171-8357613f0efa"
  ],
  "applicationType": "stb"
}
```

Response codes: 201, 400, 404, 409.

Response body: created feature rule.

Update feature rule

PUT http://<host>:<port>/featurerule

Headers: Accept = application/json Content-Type = application/json

Request body is feature rule json object

```
{
  "id": "963b2b7c-d198-4c23-820a-b56ac809c333",
  "name": "instance_test",
  "rule": {
    "negated": false,
    "condition": {
      "freeArg": {
        "type": "STRING",
        "name": "testKey"
      },
      "operation": "IS",
      "fixedArg": {
        "bean": {
          "value": {
            "java.lang.String": "testValue"
          }
        }
      }
    },
    "compoundParts": []
  },
  "priority": 1,
  "featureIds": [
    "27a9d55e-0f0c-4809-b171-8357613f0efa"
  ],
  "applicationType": "stb"
}
```

Response codes: 201, 400, 404, 409.

Response body: created feature rule.

Delete feature rule

DELETE http://<host>:<port>/featurerule/{id}

Response codes: 204, 404, 409.

Activation Minimum Version

Retrieve all activation minimum versions

GET http://<host>:<port>/amv

Headers: Accept = application/json

Response body

```
[
  {
    "id": "42670af7-6ea2-485f-9aee-1fa5895d655b",
    "applicationType": "stb",
    "description": "Activation Version",
    "regularExpressions": [],
    "model": "MX011AN",
    "firmwareVersions": [
      "FIRMWARE_VERSION"
    ]
  }
]
```

Response codes: 200.

Retrieve filtered activation minimum versions

GET http://<host>:<port>/amv/filtered?

Headers: Accept = application/json, Content-Type = application/json

Request params:

'applicationType' param is required.

1. DESCRIPTION:
'DESCRIPTION=description'
to filter feature rule by description.
2. MODEL:
'MODEL=model'
to filter by feature rule key
3. PARTNER_ID:
'PARTNER_ID=partnerId'
to filter by feature rule value
4. FIRMWARE_VERSION:
'FIRMWARE_VERSION=firmwareVersion'
to filter by feature name.
5. REGULAR_EXPRESSION:
'REGULAR_EXPRESSION=regularExpression'
6. APPLICATION_TYPE:
'APPLICATION_TYPE=stb'
to filter by application type

Parameters could be combined:

'?APPLICATION_TYPE=stb&MODEL=modelId'

Import activation version:

If activation minimum version with provided id does not exist it is imported otherwise updated.

POST http://<host>:<port>/amv/importAll

Headers: Accept = application/json Content-Type = application/json

Request body is list of activation minimum version:

```
[
  {
    "id": "42670af7-6ea2-485f-9aee-1fa5895d655b",
    "applicationType": "stb",
    "description": "Activation Version",
    "regularExpressions": [],
    "model": "MX011AN",
    "firmwareVersions": [
      "FIRMWARE_VERSION"
    ]
  }
]
```

Return list of imported and not imported activation minimum version IDs:

Response body:

```
{
  "NOT_IMPORTED": [],
  "IMPORTED": ["42670af7-6ea2-485f-9aee-1fa5895d655b"]
}
```

Response codes: 200, 400, 404, 409

Create activation minimum version

POST http://<host>:<port>/amv

Headers: Accept = application/json Content-Type = application/json

Request body is activation minimum version json object

```
{
  "id": "42670af7-6ea2-485f-9aee-1fa5895d655b",
  "applicationType": "stb",
  "description": "Activation Version",
  "regularExpressions": [],
  "model": "MX011AN",
  "firmwareVersions": [
    "FIRMWARE_VERSION"
  ]
}
```

Response codes : 201, 400, 404, 409.

Response body: created activation minimum version.

Update activation minimum version

PUT http://<host>:<port>/amv Headers: Accept = application/json Content-Type = application/json

Request body is activation minimum version json object

```
{
  "id": "42670af7-6ea2-485f-9aee-1fa5895d655b",
  "applicationType": "stb",
  "description": "Activation Version",
  "regularExpressions": [],
  "model": "MX011AN",
  "firmwareVersions": [
    "FIRMWARE_VERSION"
  ]
}
```

Response codes: 201, 400, 404, 409.

Response body: created activation minimum version.

Delete activation minimum version

DELETE http://<host>:<port>/amv/{id}

Response codes: 204, 404, 409.