Contents

Update File Cache Configuration ................................................................. 31

6 Devices .................................................................................. 33
   List Status Logs ..................................................................... 33

7 Utility ..................................................................................... 37
   Get Edge Configuration .................................................................. 37
   Get Edge Version ........................................................................ 39
   Store Authentication Token ............................................................. 40
   Determine if Authorization Token Exists ...................................... 41
The FlexNet Edge Server is installed on the customer’s network in order to forward requests to the Updates and Insights Notification Service from devices that have no direct connectivity to the Internet. End-user devices send their requests for updates to FlexNet Edge, which queues them until FlexNet Edge can connect to the Notification Service. FlexNet Edge will then carry out the requests for updates, and relay the results back to the devices once it has retrieved the required information.

The FlexNet Edge REST APIs described in this document enable you to integrate FlexNet Edge capabilities into your application or to build your own user interface for Edge.

*Note* • A set of REST services for the Updates and Insights Notification Service can be found at the following URL of your FlexNet Operations instance:

https://<siteID>.flexnetoperations.com/flexnet/swagger-ui.html#/  

These APIs are used by client applications to request and receive updates from FlexNet Edge. For more information see Getting Started with Updates and Insights.

*Security* • The Authorization header must contain the token given to you by Revenera. The header should no longer use the Basic scheme.

The FlexNet Edge Server 2021.01 API Reference Guide is organized in the following sections:

**Table 1-1** • FlexNet Edge Server 2021.01 API Reference Guide

<table>
<thead>
<tr>
<th>Topic</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base URL and Supported Ports</td>
<td>Describes the FlexNet Edge Server base URL and the supported ports.</td>
</tr>
<tr>
<td>Update Requests</td>
<td>Explains how to perform the following tasks:</td>
</tr>
<tr>
<td></td>
<td>• List Update Requests</td>
</tr>
<tr>
<td></td>
<td>• Delete Update Request</td>
</tr>
</tbody>
</table>
The following resources are available to assist you with using this product:

- Revenera Product Documentation
- Revenera Community
- Revenera Learning Center
- Revenera Support

**Revenera Product Documentation**

You can find documentation for all Revenera products on the [Revenera Product Documentation](https://www.revenera.com) site:
https://docs.revenera.com

Revenera Community

On the Revenera Community site, you can quickly find answers to your questions by searching content from other customers, product experts, and thought leaders. You can also post questions on discussion forums for experts to answer. For each of Revenera’s product solutions, you can access forums, blog posts, and knowledge base articles.

https://community.revenera.com

Revenera Learning Center

Revenera offers a variety of training courses—both instructor-led and online—to help you understand how to quickly get the most out of your Revenera products. The Revenera Learning Center offers free, self-guided, online training classes. You can also choose to participate in structured classroom training delivered as public classes. You can find a complete list of both online content and public instructor-led training in the Learning Center.

https://learning.revenera.com

Revenera Support

For customers who have purchased a maintenance contract for their product(s), you can submit a support case or check the status of an existing case by making selections on the Get Support menu of the Revenera Community.

https://community.revenera.com

Contact Us

Revenera is headquartered in Itasca, Illinois, and has offices worldwide. To contact us or to learn more about our products, visit our website at:

http://www.revenera.com

You can also follow us on social media:

- Twitter
- Facebook
- LinkedIn
- YouTube
- Instagram
This section describes the FlexNet Edge Base URL and the supported ports.

### FlexNet Edge Base URL

The REST endpoint base URL for the FlexNet Edge Server includes the server’s host name and port number in the format of `EdgeServerHostName:port`.

The host name can be either a network address (such as `myserver.example.com`) or an IP address, followed by a port number such as:

- `https://myserver.example.com:8080`
- `http://111.2.3.44:8080/`

*Note* • *Currently, the FlexNet Edge Server only supports HTTP from the client.*

### Supported Ports

The Edge Server exposes ports 8080, 8888, and 9000.

- **Requests from devices**—FlexNet Edge Server listens on port 8080 for requests from devices for updates, status log entries, device registration, and manifests.

  In the case of a blocked port or another connectivity issue that would affect the usability or setup of Edge Server, an administrator should free up port 8080 so that it can be used by the Edge Server.

- **FlexNet Edge Management Console**—Port 8888 is used only for accessing the Management Console (the FlexNet Edge user interface).
**REST Endpoints**—The REST endpoints which correspond to FlexNet Edge Management Console operations are exposed via Port 9000.

Table 2-1 • REST Endpoint URLs and Port Numbers

<table>
<thead>
<tr>
<th>Task</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Update Requests</td>
<td><a href="http://siteID.flexnetoperations.com:9000/api/v1/platform/update_requests">http://siteID.flexnetoperations.com:9000/api/v1/platform/update_requests</a></td>
</tr>
<tr>
<td>Device Groups</td>
<td><a href="http://siteID.flexnetoperations.com:9000/api/v1/platform/device_groups">http://siteID.flexnetoperations.com:9000/api/v1/platform/device_groups</a></td>
</tr>
<tr>
<td>File Cache</td>
<td><a href="http://siteID.flexnetoperations.com:9000/api/v1/platform/file_cache">http://siteID.flexnetoperations.com:9000/api/v1/platform/file_cache</a></td>
</tr>
<tr>
<td>Devices</td>
<td><a href="http://siteID.flexnetoperations.com:9000/api/v1/platform/status_logs">http://siteID.flexnetoperations.com:9000/api/v1/platform/status_logs</a></td>
</tr>
<tr>
<td>Utility</td>
<td><a href="http://siteID.flexnetoperations.com:9000/api/v1/platform/config">http://siteID.flexnetoperations.com:9000/api/v1/platform/config</a></td>
</tr>
<tr>
<td></td>
<td><a href="http://siteID.flexnetoperations.com:9000/api/v1/platform/version">http://siteID.flexnetoperations.com:9000/api/v1/platform/version</a></td>
</tr>
</tbody>
</table>
The following Update Requests API are described in this section:

- List Update Requests
- Delete Update Request

**Update Requests Request URL**

Use the following request URL to invoke Update Requests API:

http://siteID.flexnetoperations.com:9000/api/v1/platform/update_requests

**List Update Requests**

You can call the GET /update_requests API to list all cached update requests.

**API**

GET /update_requests

**Parameters**

The GET /update_requests API does not have any parameters.
Example Response

The following is an example response of the `GET /update_requests` API.

```
[{
  "hash": "ec22564afca2799",
  "update_request_created_at": "2019-02-05T15:42:35.384995Z",
  "update_request_updated_at": "2019-02-05T15:42:46.239294Z",
  "update_request_id": "5c59aeeb8479802943e9ff4d",
  "has_error": false,
  "status": "complete",
  "update_request": {
    "updateFromPkgId": "NSPref-DPRF-2",
    "language": 1033,
    "platform": "WIN32",
    "id": "Device-NSPref-AccountA-1",
    "idType": "STRING",
    "publisherName": "fnedemo"
  },
  "cached_update_items": [
    {
      "local_update_item": {
        "commandLine": null,
        "description": "Language Description for NSPref_Update-2",
        "manifestUUID": "c47e74dc-369f-4f2b-901e-780fb4e2b642",
        "expireDate": "05/01/2025",
        "fromPackageId": "NSPref-DPRF-2",
        "languageCode": 1033,
        "productName": null,
        "updateName": "NSPref_Update-2",
        "updateId": "NSPref_Update-2",
        "startDate": "11/11/2018",
        "title": "Title for NSPref_Update-2",
        "toPackageId": "NSPref-DPRT-2"
      },
      "complete": true,
      "manifest_version": "1.0",
      "manifest_files": [
        {
          "id": "5c59aefb8479802943e9ff4f",
          "createdAt": "2019-02-05T15:42:51.631023Z",
          "updatedAt": "2019-02-05T15:42:51.631023Z",
          "uniqueId": "A51DB7511D3543EFE3371445CFE13443",
          "originalName": "NSPrefUploadFile.txt",
          "contentType": "text/plain"
        }
      ]
    }
  ],
  "fno_response": {
    "updateItems": [
      {
        "commandLine": null,
        "description": "Language Description for NSPref_Update-2",
        "manifestUUID": "c47e74dc-369f-4f2b-901e-780fb4e2b642",
        "expireDate": "05/01/2025",
        "fromPackageId": "NSPref-DPRF-2",
        "languageCode": 1033,
        "productName": null,
        "updateName": "NSPref_Update-2",
        "updateId": "NSPref_Update-2",
        "startDate": "11/11/2018",
        "title": "Title for NSPref_Update-2",
        "toPackageId": "NSPref-DPRT-2"
      }
    ]
  }
}
```
Response Messages

The GET /update_requests API has the following response messages.

Table 3-1 • Response Messages

<table>
<thead>
<tr>
<th>HTTP Status Code</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>List of all cached update requests successfully returned.</td>
</tr>
</tbody>
</table>
Delete Update Request

You can use the DELETE /update_requests/{update_request_id} API to remove an update request, all its update items, and all of its manifest files from the local cache. You would do this in order to remove update items that are in error, or to free up space in the cache, or to clear out mistaken file downloads.

**API**

DELETE /update_requests/{update_request_id}

**Parameters**

The DELETE /update_requests/{update_request_id} API has the following parameter.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>update_request_id</td>
<td>String</td>
<td>Use this parameter to specify the update request to delete. This value is obtained from the GET /update_requests API.</td>
</tr>
</tbody>
</table>

*Note* • Parameters marked with an asterisk (*) are required.

**Response Messages**

The DELETE /update_requests/{update_request_id} API has the following response messages.

<table>
<thead>
<tr>
<th>HTTP Status Code</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>204</td>
<td>The update request was successfully deleted.</td>
</tr>
</tbody>
</table>
The following Device Groups API are described in this section:

- Get All Device Groups
- Create a Device Group
- Update a Device Group by ID
- Delete a Device Group by ID
- Get a Device Group by ID
- Get All Device IDs for a Device Group
- Set (Replace) the Device IDs on a Device Group
- Add Device ID to Device Group
- Remove Device ID from a Device Group

**Device Groups Request URL**

Use the following request URL to invoke Device Groups API:

http://siteID.flexnetoperations.com:9000/api/v1/platform/device_groups
Get All Device Groups

You can use the GET /device_groups API to return all device groups.

**API**

GET /device_groups

**Parameters**

The GET /device_groups API has the following parameters.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>deviceID</td>
<td>String</td>
<td>Use this optional parameter to filter the list of groups to those that contain the specified device ID.</td>
</tr>
</tbody>
</table>

**Example Body**

The following is an example body of a device group.

```json
{
  "name": "string",
  "activeState": "string",
  "installWindow": {
    "openWeekdays": [
      0
    ],
    "openWeeksDivisibleBy": 0
  },
  "deviceIds": [
    "string"
  ]
}
```

**Response Messages**

The GET /device_groups API has the following response messages.

<table>
<thead>
<tr>
<th>HTTP Status Code</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>List of device groups is successfully returned.</td>
</tr>
</tbody>
</table>
Create a Device Group

You can use the **POST /device_groups** API to create a device group.

### API

**POST /device_groups**

### Parameters

The **POST /device_groups** API has the following parameters.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>deviceGroup</td>
<td>name</td>
<td>String</td>
<td>Name of device group.</td>
</tr>
<tr>
<td>activeState</td>
<td>String</td>
<td></td>
<td>Set to one of the following values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- <strong>inactive</strong>—Never allow installs for devices in group.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- <strong>active</strong>—Always allow installs for devices in group.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- <strong>as_scheduled</strong>—Allow installs per <strong>installWindow</strong>.</td>
</tr>
<tr>
<td>installWindow</td>
<td>String</td>
<td></td>
<td>Defines the device group’s installation window using the following values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- <strong>openWeekdays</strong>—Must have an array of 0-6 open days, with no duplicates,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>where each number is between 0 and 6.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- <strong>openWeeksDivisibleBy</strong>—Must be between 1 and 2, interpreted as <strong>Every</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Week and <strong>Every Other Week</strong>, respectively, by the user interface.</td>
</tr>
</tbody>
</table>

**Note** • When evaluated by Edge, the current time stamp’s ISO Week ([https://en.wikipedia.org/wiki/ISO_week_date](https://en.wikipedia.org/wiki/ISO_week_date)) is divided by **openWeeksDivisibleBy**, and if the remainder is not 0, the install is not allowed.

| deviceIds        | String | List of device IDs.          |

**Note** • Parameters marked with an asterisk (*) are required.
Example Body

The following is an example body of a device group.

```
{
    "name": "string",
    "activeState": "string",
    "installWindow": {
        "openWeekdays": [
            0
        ],
        "openWeeksDivisibleBy": 0
    },
    "deviceIds": [
        "string"
    ]
}
```

Response Messages

The POST /device_groups API has the following response messages.

**Table 4-4 • Response Messages**

<table>
<thead>
<tr>
<th>HTTP Status Code</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>201</td>
<td>Device group has been successfully created. The response shows the body of the newly created device group.</td>
</tr>
<tr>
<td>400</td>
<td>Error indicating that one of the following has occurred:</td>
</tr>
<tr>
<td></td>
<td>● Device group must have a name and an install window.</td>
</tr>
<tr>
<td></td>
<td>● Device IDs may not belong to other groups.</td>
</tr>
<tr>
<td></td>
<td>● Install windows must have an array of 0-6 days, with no duplicates, where each number is between 0 and 6.</td>
</tr>
<tr>
<td></td>
<td>● Install windows <strong>openWeeksDivisibleBy</strong> must be between 1 and 2.</td>
</tr>
</tbody>
</table>
Update a Device Group by ID

You can use the `GET /device_groups/{id}` API to update a device group by ID.

**API**

**PUT /device_groups/{id}**

**Parameters**

The `PUT /device_groups/{id}` API has the following parameters.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>deviceGroup</code> *</td>
<td>name</td>
<td>String</td>
<td>Name of device group.</td>
</tr>
<tr>
<td><code>activeState</code></td>
<td></td>
<td>String</td>
<td>Set to one of the following values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• <code>inactive</code>—Never allow installs for devices in group.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• <code>active</code>—Always allow installs for devices in group.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• <code>as_scheduled</code>—Allow installs per <code>installWindow</code>.</td>
</tr>
<tr>
<td><code>installWindow</code></td>
<td></td>
<td>String</td>
<td>Defines the device group’s installation window using the following values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• <code>openWeekdays</code>—Must have an array of 0-6 open days, with no duplicates,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>where each number is between 0 and 6.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• <code>openWeeksDivisibleBy</code>—Must be between 1 and 2,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>interpreted as <code>Every Week</code> and <code>Every Other Week</code>, respectively, by the</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>user interface.</td>
</tr>
<tr>
<td><code>deviceIds</code></td>
<td></td>
<td>String</td>
<td>List of device IDs.</td>
</tr>
<tr>
<td><code>id</code> *</td>
<td></td>
<td>String</td>
<td>Device group ID.</td>
</tr>
</tbody>
</table>

*Note* • Parameters marked with an asterisk (*) are required.

*Note* • When evaluated by Edge, the current time stamp’s ISO Week ([https://en.wikipedia.org/wiki/ISO_week_date](https://en.wikipedia.org/wiki/ISO_week_date)) is divided by `openWeeksDivisibleBy`, and if the remainder is not 0, the install is not allowed.
Example Response

The following is an example body of a device group.

{  
  "name": "string",
  "activeState": "string",
  "installWindow": {
    "openWeekdays": [0],
    "openWeeksDivisibleBy": 0
  },
  "deviceIds": [
    "string"
  ]
}

Response Messages

The PUT /device_groups/{id} API has the following response messages.

Table 4-6 • Response Messages

<table>
<thead>
<tr>
<th>HTTP Status Code</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>OK. Update has been successfully performed.</td>
</tr>
<tr>
<td>400</td>
<td>Error indicating that one of the following has occurred:</td>
</tr>
<tr>
<td></td>
<td>● Device group must have a name and an install window.</td>
</tr>
<tr>
<td></td>
<td>● Device IDs may not belong to other groups.</td>
</tr>
<tr>
<td></td>
<td>● Install windows must have an array of 0-6 open days, with no duplicates, where each number is between 0 and 6.</td>
</tr>
<tr>
<td></td>
<td>● Install windows <code>openWeeksDivisibleBy</code> must be between 1 and 2.</td>
</tr>
</tbody>
</table>
Delete a Device Group by ID

You can use the DELETE /device_groups/{id} API to delete a device group by ID.

**API**

DELETE /device_groups/{id}

**Parameters**

The DELETE /device_groups/{id} API has the following parameters.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>String</td>
<td>Device group ID.</td>
</tr>
</tbody>
</table>

*Note* • Parameters marked with an asterisk (*) are required.

**Response Messages**

The DELETE /device_groups/{id} API has the following response messages.

<table>
<thead>
<tr>
<th>HTTP Status Code</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>204</td>
<td>The device group was successfully deleted.</td>
</tr>
</tbody>
</table>
Get a Device Group by ID

You can use the GET /device_groups/{id} API to return a specified device group.

**API**

GET /device_groups/{id}

**Parameters**

The GET /device_groups/{id} API has the following parameters.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id *</td>
<td>String</td>
<td>Device group ID.</td>
</tr>
</tbody>
</table>

**Note** • Parameters marked with an asterisk (*) are required.

**Example Body**

The following is an example body of a device group.

```json
{
   "name": "string",
   "activeState": "string",
   "installWindow": {
      "openWeekdays": [0],
      "openWeeksDivisibleBy": 0,
   },
   "deviceIds": ["string"
}
```

**Response Messages**

The GET /device_groups/{id} API has the following response messages.

<table>
<thead>
<tr>
<th>HTTP Status Code</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>A device group is returned.</td>
</tr>
</tbody>
</table>
Get All Device IDs for a Device Group

You can use the **GET/device_groups/{id}/devices** API to return a list of all device IDs for a device group.

**API**

GET /device_groups/{id}/devices

**Parameters**

The **GET /device_groups/{id}/devices** API has the following parameters.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id *</td>
<td>String</td>
<td>Device group ID.</td>
</tr>
</tbody>
</table>

**Note** - Parameters marked with an asterisk (*) are required.

**Example Response**

```
[  
  "string"
]
```

**Response Messages**

The **GET /device_groups/{id}/devices** API has the following response messages.

<table>
<thead>
<tr>
<th>HTTP Status Code</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>A list of the specified device group’s device IDs are returned.</td>
</tr>
</tbody>
</table>
Set (Replace) the Device IDs on a Device Group

You can use the **PUT /device_groups/{id}/devices** API to set (replace) the device IDs of a specified device group.

**API**

**PUT /device_groups/{id}/devices**

**Parameters**

The **PUT /device_groups/{id}/devices** API has the following parameters.

**Table 4-13 • Parameters**

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id *</td>
<td>String</td>
<td>Device group ID.</td>
</tr>
<tr>
<td>deviceIds *</td>
<td>Array [String]</td>
<td>List of device IDs.</td>
</tr>
</tbody>
</table>

**Note**  • Parameters marked with an asterisk (*) are required.

**Response Messages**

The **PUT /device_groups/{id}/devices** API has the following response messages.

**Table 4-14 • Response Messages**

<table>
<thead>
<tr>
<th>HTTP Status Code</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>OK. The device IDs of the specified device group have been successfully replaced.</td>
</tr>
<tr>
<td>400</td>
<td>Device ID belongs to another group. Device IDs are not permitted to be added to more than one device group.</td>
</tr>
</tbody>
</table>
Add Device ID to Device Group

You can use the `PUT /device_groups/{id}/devices/{device_id}` API to add a device ID to a device group.

**API**

`PUT /device_groups/{id}/devices/{device_id}`

**Parameters**

The `PUT /device_groups/{id}/devices/{device_id}` API has the following parameters.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>String</td>
<td>Device group ID.</td>
</tr>
<tr>
<td>device_id</td>
<td>String</td>
<td>Device ID.</td>
</tr>
</tbody>
</table>

*Note: Parameters marked with an asterisk (*) are required.*

**Response Messages**

The `PUT /device_groups/{id}/devices/{device_id}` API has the following response messages.

<table>
<thead>
<tr>
<th>HTTP Status Code</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>204</td>
<td>The device ID was successfully added to the device group.</td>
</tr>
</tbody>
</table>
Remove Device ID from a Device Group

You can use the `DELETE /device_groups/{id}/devices/{device_id}` API to remove a device ID from a device group.

**API**

```
DELETE /device_groups/{id}/devices/{device_id}
```

**Parameters**

The `DELETE /device_groups/{id}/devices/{device_id}` API has the following parameters.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id *</td>
<td>String</td>
<td>Device group ID.</td>
</tr>
<tr>
<td>device_id *</td>
<td>String</td>
<td>Device ID.</td>
</tr>
</tbody>
</table>

**Note** • Parameters marked with an asterisk (*) are required.

**Response Messages**

The `DELETE /device_groups/{id}/devices/{device_id}` API has the following response messages.

<table>
<thead>
<tr>
<th>HTTP Status Code</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>204</td>
<td>The device ID was successfully removed from the device group.</td>
</tr>
</tbody>
</table>
The following File Cache API are described in this section:

- List File Cache
- Clear File Cache
- Get File Cache Configuration
- Update File Cache Configuration

**File Cache Request URL**

Use the following request URL to invoke File Cache API:

```
http://siteID.flexnetoperations.com:9000/api/v1/platform/file_cache
```
List File Cache

You can use the GET /file_cache API to request the current contents of the file cache.

API

GET /file_cache

Parameters

The GET /file_cache API has the following parameters.

Table 5-1 • Parameters

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>accept</td>
<td>String</td>
<td>The default value is application/json.</td>
</tr>
</tbody>
</table>

Example Response

The following is an example response of the GET /file_cache API.

```
{
    "files": [
        {
            "name": "string",
            "size": 0,
            "sizeHuman": "string"
        }
    ],
    "totalBytes": "string",
    "totalBytesHuman": "string"
}
```

Response Messages

The GET /file_cache API has the following response messages.

Table 5-2 • Response Messages

<table>
<thead>
<tr>
<th>HTTP Status Code</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>The current contents of the file cache were returned successfully.</td>
</tr>
</tbody>
</table>
Clear File Cache

You can use the DELETE /file_cache/clear API to purge all cached updated request records and their associated files.

API

DELETE /file_cache/clear

Parameters

The DELETE /file_cache/clear API has the following parameters.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>accept *</td>
<td>String</td>
<td>The default value is application/json.</td>
</tr>
<tr>
<td>content-type *</td>
<td>String</td>
<td>The default value is application/json.</td>
</tr>
</tbody>
</table>

Note • Parameters marked with an asterisk (*) are required.

Response Messages

The DELETE /file_cache/clear API has the following response messages.

<table>
<thead>
<tr>
<th>HTTP Status Code</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>204</td>
<td>All cached updated request records and their associated files have been successfully purged.</td>
</tr>
</tbody>
</table>
Get File Cache Configuration

You can use the GET /file_cache/config API to return the current configuration of the file cache. This call relates to the file cache configuration in particular, but the configuration is also available in the /config call.

**API**

GET /file_cache/config

**Parameters**

The GET /file_cache/config API has the following parameters.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>accept *</td>
<td>String</td>
<td>The default value is application/json.</td>
</tr>
</tbody>
</table>

**Example Response**

The following is an example response of the GET /file_cache/config API.

```json
{
    "ttl_seconds": 0
}
```

**Response Messages**

The GET /file_cache/config API has the following response messages.

<table>
<thead>
<tr>
<th>HTTP Status Code</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>The current configuration of the file cache has been successfully returned.</td>
</tr>
</tbody>
</table>
Update File Cache Configuration

You can use the `PUT /file_cache/config` API to change the cache configuration.

**API**

`PUT /file_cache/config`

**Parameters**

The `PUT /file_cache/config` API has the following parameters.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>accept</td>
<td>String</td>
<td><strong>application/json</strong>.</td>
<td></td>
</tr>
<tr>
<td>content-type</td>
<td>String</td>
<td><strong>application/json</strong>.</td>
<td></td>
</tr>
<tr>
<td>body</td>
<td><code>ttl_seconds</code></td>
<td>Object</td>
<td>Number ($double).</td>
</tr>
</tbody>
</table>

*Note*: Parameters marked with an asterisk (*) are required.

**Example Body**

The following is an example body of a `PUT /file_cache/config` API.

```
{
  "ttl_seconds": 0
}
```

**Response Messages**

The `PUT /file_cache/config` API has the following response messages.

<table>
<thead>
<tr>
<th>HTTP Status Code</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>204</td>
<td>The cache configuration has been successfully changed.</td>
</tr>
</tbody>
</table>
Chapter 5  File Cache
Update File Cache Configuration
The following Devices API are described in this section:

- List Status Logs

**Devices Request URL**

Use the following request URL to invoke Devices API:

http://siteID.flexnetoperations.com:9000/api/v1/platform/status_logs

**List Status Logs**

You can use the **GET /status_logs** API to get the most recent status logs, grouped by device.

**API**

GET /status_logs

**Parameters**

The **GET /status_logs** API has the following parameters.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>accept</td>
<td>String</td>
<td>The default value is application/json.</td>
</tr>
</tbody>
</table>

*Note* • Parameters marked with an asterisk (*) are required.
Example Response

The following is an example response of the **GET /status_logs** API.

```json
[
  {
    "origin": "string",
    "created_at": "string",
    "updated_at": "string",
    "id": "string",
    "idType": "string",
    "language": 0,
    "packageId": "string",
    "publisherName": "string",
    "platform": "string",
    "statusCode": 0,
    "updateId": "string",
    "updateName": "string",
    "timeStamp": "string"
  }
]
```

Example Response Values

The example **GET /status_logs** API response has the following values.

<table>
<thead>
<tr>
<th>Values</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>origin</td>
<td>String</td>
<td>Unused. May be set to 'edge' or ignored.</td>
</tr>
<tr>
<td>created_at</td>
<td>String</td>
<td>Date and time status log was created.</td>
</tr>
<tr>
<td>updated_at</td>
<td>String</td>
<td>Date and time status log was updated.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>ID of the device.</td>
</tr>
<tr>
<td>idType</td>
<td>String</td>
<td>ID type of the device.</td>
</tr>
<tr>
<td>language</td>
<td>Number ($double)</td>
<td>Number indicating the language of the update.</td>
</tr>
<tr>
<td>packageId</td>
<td>String</td>
<td>Package ID.</td>
</tr>
<tr>
<td>publisherName</td>
<td>String</td>
<td>Name of publisher.</td>
</tr>
<tr>
<td>platform</td>
<td>String</td>
<td>Platform</td>
</tr>
<tr>
<td>statusCode</td>
<td>Number ($double)</td>
<td>Status code.</td>
</tr>
<tr>
<td>updateId</td>
<td>String</td>
<td>Update ID.</td>
</tr>
<tr>
<td>updateName</td>
<td>String</td>
<td>Update name.</td>
</tr>
</tbody>
</table>
Table 6-2 • Values

<table>
<thead>
<tr>
<th>Values</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>timeStamp</td>
<td>String</td>
<td>Date and time of the update.</td>
</tr>
</tbody>
</table>

Response Messages

The GET /status_logs API has the following response messages.

Table 6-3 • Response Messages

<table>
<thead>
<tr>
<th>HTTP Status Code</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>Status logs, grouped by device, are successfully returned.</td>
</tr>
</tbody>
</table>
The following Utility API are described in this section:

- Get Edge Configuration
- Get Edge Version
- Store Authentication Token
- Determine if Authorization Token Exists

Utility Request URLs

Use the following request URLs to invoke Utility API:

- `http://siteID.flexnetoperations.com:9000/api/v1/platform/config`
- `http://siteID.flexnetoperations.com:9000/api/v1/platform/version`

**Get Edge Configuration**

You can use the `GET /config` API to request the current configuration of the Edge Server.

**API**

`GET /config`

**Parameters**

The `GET /config` API has the following parameters.

**Table 7-1 • Parameters**

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>accept</td>
<td>String</td>
<td>Default value is <code>application/json</code>.</td>
</tr>
</tbody>
</table>
Note • Parameters marked with an asterisk (*) are required.

Example Response

The following is an example response of the GET /config API.

```
{
    "fno_protocol": "string",
    "fno_host": "string",
    "fno_port": "string",
    "fno_base_url": "string",
    "status_log_url": "string",
    "update_request_url": "string",
    "pause_update_delivery": true,
    "manifest_filestore_path": "string",
    "manifest_filestore_ttl_seconds": 0,
    "manifest_filestore_purge_job_queue_config": {
        "max_queue_size": 0,
        "burst_buffer_size": 0,
        "num_workers": 0,
        "rate_limit_seconds": 0
    },
    "update_request_job_queue_config": {
        "max_queue_size": 0,
        "burst_buffer_size": 0,
        "num_workers": 0,
        "rate_limit_seconds": 0
    },
    "status_log_job_queue_config": {
        "max_queue_size": 0,
        "burst_buffer_size": 0,
        "num_workers": 0,
        "rate_limit_seconds": 0
    },
    "update_item_job_queue_config": {
        "max_queue_size": 0,
        "burst_buffer_size": 0,
        "num_workers": 0,
        "rate_limit_seconds": 0
    }
}
```

Response Messages

The GET /config API has the following response messages.

Table 7-2 • Response Messages

<table>
<thead>
<tr>
<th>HTTP Status Code</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>A JSON body containing the FlexNet Edge Server configuration is returned.</td>
</tr>
</tbody>
</table>
Get Edge Version

You can use the **GET /version** API to request the current version of FlexNet Edge.

**API**

GET /version

**Parameters**

The **GET /version** API has the following parameters.

**Table 7-3 • Parameters**

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>accept *</td>
<td>String</td>
<td>Default value is <code>application/json</code>.</td>
</tr>
</tbody>
</table>

**Note** • Parameters marked with an asterisk (*) are required.

**Example Response**

The following is an example response of the **GET /version** API.

```
{
    "version": "1.12-flexera"
}
```

**Response Messages**

The **GET /version** API has the following response messages.

**Table 7-4 • Response Messages**

<table>
<thead>
<tr>
<th>HTTP Status Code</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>A JSON body containing the current version of FlexNet Edge is successfully returned.</td>
</tr>
</tbody>
</table>
Store Authentication Token

You can use the **PUT /authorization-token** API to store the jwt authorization token to be included on all update requests.

**API**

**PUT /authorization-token**

**Parameters**

The **PUT /authorization-token** API has the following parameters.

**Table 7-5 • Parameters**

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>accept *</td>
<td></td>
<td>String</td>
<td>Default value is application/json.</td>
</tr>
<tr>
<td>body *</td>
<td>token</td>
<td>Object</td>
<td>String</td>
</tr>
</tbody>
</table>

**Note** • Parameters marked with an asterisk (*) are required.

**Example Value**

The following is an example value of the **PUT /authorization-token** API.

```json
{
    "token": "jwtcontents"
}
```

**Authorization Token Details**

The Edge Server authorization token is required to be in the header, which is saved in the following file:

```
/opt/edge/conf/www_flexera/api-config.service.js
```

The Flexera demo authorization token is:

```javascript
const flexera_demo_auth_token = 'B1Je4.rC6UW0tQ7fRpnf1plAbV'
```

Every request against the private URL must include the following headers:

```javascript
{
    'Accept': 'application/json',
    'Content-Type': 'application/json',
    'Authorization': 'B1Je4.rC6UW0tQ7fRpnf1plAbV'
}
```

**Note** • For further information about JSON Web Tokens, refer to [https://tools.ietf.org/html/rfc7515](https://tools.ietf.org/html/rfc7515). The documentation for your JWS/JWT library will describe how to create JWTs programmatically.
Response Messages

The PUT /authorization-token API has the following response messages.

Table 7-6 • Response Messages

<table>
<thead>
<tr>
<th>HTTP Status Code</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>204</td>
<td>The server successfully processed the request but is not returning any content.</td>
</tr>
</tbody>
</table>

Determine if Authorization Token Exists

You can use the GET /authorization-token/exists API to determine whether the authorization token exists.

API

GET /authorization-token/exists

Parameters

The GET /authorization-token/exists API has the following parameters.

Table 7-7 • Parameters

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>accept *</td>
<td>String</td>
<td>Default value is application/json.</td>
</tr>
</tbody>
</table>

Note • Parameters marked with an asterisk (*) are required.

Example Response

The following is an example response of the GET /authorization-token/exists API.

```json
{ 
  "token_exists": true 
}
```

Response Messages

The GET /authorization-token/exist API has the following response messages.

Table 7-8 • Response Messages

<table>
<thead>
<tr>
<th>HTTP Status Code</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>Boolean response (true or false) returned.</td>
</tr>
</tbody>
</table>
Chapter 7  Utility

Determine if Authorization Token Exists