Synchronoss

Digital Vault Extensions

API Developer Guide

Version 2.14.1

March 2015

CONFIDENTIAL

Synchronoss

Digital Vault Extensions

Product Version: 2.14.1

Publication Date: 2015-03-02 05:53:45

All product names, company names and/or trademarks mentioned in this Document remain the property of their respective owners. This Document or any software or computer code delivered by Synchronoss Technologies, Inc. and its affiliates ("Synchronoss") is confidential to Synchronoss and is provided by Synchronoss directly to the recipient on the strict understanding that no part of this document or any software or computer code delivered by Synchronoss may be modified, translated, decompiled, disassembled, reproduced, reverse engineered, used or transmitted by the recipient in any form without the prior written consent of Synchronoss.

THE INFORMATION PROVIDED IN THIS DOCUMENT OR ANY SOFTWARE OR COMPUTER CODE DELIVERED BY SYNCHRONOSS IS FOR INFORMATION AND REVIEW PURPOSES ONLY AND SHALL NOT IN ANY CIRCUMSTANCES CONSTITUTE OR BE DEEMED TO CONSTITUTE A WARRANTY BY SYNCHRONOSS AS TO THE ACCURACY OF SUCH INFORMATION, ADVICE, STATEMENTS OR RECOMMENDATIONS OR A PERFORMANCE WARRANTY WITH RESPECT TO ANY SOFTWARE OR COMPUTER CODE DELIVERED BY SYNCHRONOSS. NO BINDING LEGAL COMMITMENT SHALL BE DEEMED TO BE IN EXISTENCE UNTIL SUCH TIME AS FORMAL LEGAL AGREEMENTS ARE ENTERED INTO BY SYNCHRONOSS AND SYNCHRONOSS SHALL HAVE NO LIABILITY WHATSOEVER IN RELATION TO THIS DOCUMENT OR ANY SOFTWARE OR COMPUTER CODE DELIVERED BY SYNCHRONOSS AND SHALL NOT BE LIABLE FOR ANY LOSS, EXPENSE, DAMAGE OR CLAIM ARISING OUT OF THE ADVICE GIVEN OR NOT GIVEN OR STATEMENTS MADE OR OMITTED TO BE MADE IN CONNECTION WITH THIS DOCUMENT.

Table of Contents

1

File APIs

Digital Vault Extensions provides the following file creation APIs for use by web clients only.

1.1. dvx.user.repository.file.bulk.create

Description

Uploads multiple files using multi-part form data. Should only be used by web clients that have limited capability in terms of the data they can send.

HTTP Method

POST

URI Structure

```
/user/{userUid}/repository/{repository}/file/bulk
```

Table 1.1. URI Variables

Name	Description
`userUid`	The user's unique ID.
`repository`	The name of the user's repository.

Request Headers

Calls to this API can include the following headers. Others headers may be required on a per project basis.

Table 1.2. Request Headers

Name	Requirement	Multiple	Type	Description
`Content-Type`	Mandatory	No	string	The following values are supported: • `multipart/form-data`
`Accept`	Optional	No	string	The following values are supported: • `application/vnd.newbay.dv-1.13+json` (default) • `application/vnd.newbay.dv-1.13+xml` • `application/vnd.newbay.dv-1.13+pbuf`

Request Parameters

This API has no request parameters.

Response Codes

Successful responses return the following status codes:

- `201`

Failed operations return the following status codes:

- `400 401 403 404 406 415 500 502 503 504`

Supported Response Formats

On success, this API returns an object of type `files`, as defined in the core Digital Vault API model.

- `application/vnd.newbay.dv-1.13+json`
- `application/vnd.newbay.dv-1.13+xml`
- `application/vnd.newbay.dv-1.13+pbuf`

Errors are returned in the following formats:

- `application/vnd.newbay.dv-1.13+json`
- `application/vnd.newbay.dv-1.13+xml`
- `application/vnd.newbay.dv-1.13+pbuf`

Sample Request and Response

Example 1.1. Request

```
POST /user/48dfe6f4ebf7433a962170ffac3780d7/repository/desktop/file/bulk
Authorization: NWB token=XXXXXXXXXXXXXXXX
Host: localhost:19006
Content-Length: 59042
User-Agent: Jakarta Commons-HttpClient/3.1
Content-Type: multipart/form-data; boundary=3TKxXvb8ZmWu9KEaTKui1RFR15ZKU-4aCpMJ
Accept: application/vnd.newbay.dv-1.14+xml

--3TKxXvb8ZmWu9KEaTKui1RFR15ZKU-4aCpMJ
Content-Disposition: form-data; name="path"
Content-Type: text/plain; charset=US-ASCII
Content-Transfer-Encoding: binary

/images
--3TKxXvb8ZmWu9KEaTKui1RFR15ZKU-4aCpMJ
Content-Disposition: form-data; name="FileCount"
Content-Type: text/plain; charset=US-ASCII
Content-Transfer-Encoding: binary

2
--3TKxXvb8ZmWu9KEaTKui1RFR15ZKU-4aCpMJ
Content-Disposition: form-data; name="FileName_1"
Content-Type: text/plain; charset=US-ASCII
Content-Transfer-Encoding: binary

jet.jpg
--3TKxXvb8ZmWu9KEaTKui1RFR15ZKU-4aCpMJ
Content-Disposition: form-data; name="SourceFile_1"; filename="dve-plane2.jpg"
Content-Type: image/jpeg; charset=UTF-8
Content-Transfer-Encoding: binary

[binary stream - omitted]
--3TKxXvb8ZmWu9KEaTKui1RFR15ZKU-4aCpMJ
Content-Disposition: form-data; name="FileName_2"
Content-Type: text/plain; charset=US-ASCII
Content-Transfer-Encoding: binary

plane.jpg
--3TKxXvb8ZmWu9KEaTKui1RFR15ZKU-4aCpMJ
```

```
Content-Disposition: form-data; name="SourceFile_2"; filename="dve-plane.jpg"
Content-Type: image/jpeg; charset=UTF-8
Content-Transfer-Encoding: binary

[binary stream - omitted]
--3TKxXvb8ZmWu9KEaTKui1RFR15ZKU-4aCpMJ
```

Example 1.2. Response

```
HTTP/1.1 200 OK
content-type: application/vnd.newbay.dv-1.14+xml;charset=UTF-8

<?xml version="1.0" encoding="UTF-8"?>
<files xmlns="http://dv.newbay.com/ns/1.0"
  xmlns:a="http://alternate.newbay.com/ns/1.0"
  xmlns:dvi="http://internal.dv.newbay.com/ns/1.0" xmlns:xspf="http://xspf.org/ns/0/">
  <file>
    <link rel="file">http://localhost:19005/api/user/48dfe6f4ebf7433a962170ffac3780d7/repository/
desktop/file?path=%2Fimages%2Fjet.jpg</link>
    <link rel="content">http://localhost:19005/api/user/48dfe6f4ebf7433a962170ffac3780d7/repository/
desktop/file/content?path=%2Fimages%2Fjet.jpg</link>
    <name>jet.jpg</name>
    <parentPath>/images</parentPath>
    <author>
      <phone>11111111001</phone>
      <email>dv_functional_test_alice001@synchronoss.com</email>
      <firstName>Alice</firstName>
      <lastName>Joyce</lastName>
    </author>
    <size>19374</size>
    <versionCreated>2015-03-02T10:53:09.776Z</versionCreated>
    <systemAttribute name="Capture-Date">2008-12-22T16:27:11.000Z</systemAttribute>
    <systemAttribute name="Height">160</systemAttribute>
    <systemAttribute name="Mime-Type">image/jpeg</systemAttribute>
    <systemAttribute name="Timeline-Date">2008-12-22T16:27:11.000Z</systemAttribute>
    <systemAttribute name="Width">160</systemAttribute>
    <changes>
      <change type="create">
        <detail name="path">/images/jet.jpg</detail>
      </change>
    </changes>
    <attributesChanged>true</attributesChanged>
    <deleted>false</deleted>
    <repository>desktop</repository>
    <uri>dv-file://48dfe6f4ebf7433a962170ffac3780d7:desktop/images/jet.jpg</uri>
    <commentCount>0</commentCount>
    <version>1</version>
    <checksum>1252a9ba0460b402f1a1e32eca887e400bc44cc113de9fc268728c8b2278e824</checksum>
    <extension>jpg</extension>

  <contentToken>AADRNS5vaa0geFfKGWib_SL6Y6XS5OfSiVGAV58SqcoW34fXZti5dRGlea8mh1oGiwDiiNcBtoLoCVxqoTeeT_k~</
contentToken>
    <contentChanged>true</contentChanged>
  </file>
  <file>
    <link rel="file">http://localhost:19005/api/user/48dfe6f4ebf7433a962170ffac3780d7/repository/
desktop/file?path=%2Fimages%2Fplane.jpg</link>
    <link rel="content">http://localhost:19005/api/user/48dfe6f4ebf7433a962170ffac3780d7/repository/
desktop/file/content?path=%2Fimages%2Fplane.jpg</link>
    <name>plane.jpg</name>
    <parentPath>/images</parentPath>
    <author>
      <phone>11111111001</phone>
      <email>dv_functional_test_alice001@synchronoss.com</email>
      <firstName>Alice</firstName>
      <lastName>Joyce</lastName>
    </author>
    <size>38592</size>
    <versionCreated>2015-03-02T10:53:09.776Z</versionCreated>
    <systemAttribute name="Capture-Date">2007-02-08T17:54:23.000Z</systemAttribute>
    <systemAttribute name="Height">160</systemAttribute>
    <systemAttribute name="Mime-Type">image/jpeg</systemAttribute>
    <systemAttribute name="Timeline-Date">2007-02-08T17:54:23.000Z</systemAttribute>
    <systemAttribute name="Width">160</systemAttribute>
    <changes>
      <change type="create">
        <detail name="path">/images/plane.jpg</detail>
      </change>
    </changes>
    <attributesChanged>true</attributesChanged>
    <deleted>false</deleted>
    <repository>desktop</repository>
    <uri>dv-file://48dfe6f4ebf7433a962170ffac3780d7:desktop/images/plane.jpg</uri>
    <commentCount>0</commentCount>
    <version>1</version>
    <checksum>fc1624ff53e9745bd64df7fdadbc7d864413d2743c0cc36d56a3199b806a5b1a</checksum>
    <extension>jpg</extension>

  <contentToken>AIJ2HrqDg1TiGaSpz2cNSz0f0dT8_GwDEvxY8UBNWOaE9xRkdlXcvl68Gw0cV0rO89ZUPDRFaC08G2pS00IBG_s~</
contentToken>
    <contentChanged>true</contentChanged>
  </file>
</files>
```

1.2. dvx.user.repository.file.single.create

Description

Uploads a file using multi-part form data. Should only be used by web clients that have limited capability in terms of the data they can send.

HTTP Method

POST

URI Structure

`/user/{userUid}/repository/{repository}/file/single`

Table 1.3. URI Variables

Name	Description
`userUid`	The user's unique ID.
`repository`	The name of the user's repository.

Request Headers

Calls to this API can include the following headers. Others headers may be required on a per project basis.

Table 1.4. Request Headers

Name	Requirement	Multiple	Type	Description
`Content-Type`	Mandatory	No	string	The following values are supported: • `multipart/form-data`

Name	Requirement	Multiple	Type	Description
`Accept`	Optional	No	string	The following values are supported: • `application/vnd.newbay.dv-1.13+json` (default) • `application/vnd.newbay.dv-1.13+xml` • `application/vnd.newbay.dv-1.13+pbuf`

Request Parameters

This API has no request parameters.

Response Codes

Successful responses return the following status codes:

- `201`

Failed operations return the following status codes:

- `400 401 403 404 406 415 500 502 503 504`

Supported Response Formats

On success, this API returns an object of type `files`, as defined in the core Digital Vault API model.

- `application/vnd.newbay.dv-1.13+json`
- `application/vnd.newbay.dv-1.13+xml`
- `application/vnd.newbay.dv-1.13+pbuf`

Errors are returned in the following formats:

- `application/vnd.newbay.dv-1.13+json`
- `application/vnd.newbay.dv-1.13+xml`
- `application/vnd.newbay.dv-1.13+pbuf`

Sample Request and Response

Example 1.3. Request

```
POST /user/6d20706ee27c447495d16642f37995c8/repository/desktop/file/single
Authorization: NWB token=XXXXXXXXXXXXXXXX
Host: localhost:19006
Content-Length: 39132
User-Agent: Jakarta Commons-HttpClient/3.1
Content-Type: multipart/form-data; boundary=xUlcz-q-Jso9O3mbObHnoZio6njN2yV
Accept: application/vnd.newbay.dv-1.14+xml

--xUlcz-q-Jso9O3mbObHnoZio6njN2yV
Content-Disposition: form-data; name="path"
```

```
Content-Type: text/plain; charset=US-ASCII
Content-Transfer-Encoding: binary

/images
--xUlcz-q-Jso9O3mbObHnoZio6njN2yV
Content-Disposition: form-data; name="name"
Content-Type: text/plain; charset=US-ASCII
Content-Transfer-Encoding: binary

plane.jpg
--xUlcz-q-Jso9O3mbObHnoZio6njN2yV
Content-Disposition: form-data; name="file"; filename="dve-plane.jpg"
Content-Type: application/octet-stream; charset=UTF-8
Content-Transfer-Encoding: binary

[binary stream - omitted]
--xUlcz-q-Jso9O3mbObHnoZio6njN2yV
```

Example 1.4. Response

```
HTTP/1.1 200 OK
content-type: application/vnd.newbay.dv-1.14+xml;charset=UTF-8

<?xml version="1.0" encoding="UTF-8"?>
<files xmlns="http://dv.newbay.com/ns/1.0"
  xmlns:a="http://alternate.newbay.com/ns/1.0"
  xmlns:dvi="http://internal.dv.newbay.com/ns/1.0" xmlns:xspf="http://xspf.org/ns/0/">
  <file>
    <link rel="file">http://localhost:19005/api/user/6d20706ee27c447495d16642f37995c8/repository/
desktop/file?path=%2Fimages%2Fplane.jpg</link>
    <link rel="content">http://localhost:19005/api/user/6d20706ee27c447495d16642f37995c8/repository/
desktop/file/content?path=%2Fimages%2Fplane.jpg</link>
    <name>plane.jpg</name>
    <parentPath>/images</parentPath>
    <author>
      <phone>11111111001</phone>
      <email>dv_functional_test_alice001@synchronoss.com</email>
      <firstName>Alice</firstName>
      <lastName>Joyce</lastName>
    </author>
    <size>38592</size>
    <versionCreated>2015-03-02T10:53:10.436Z</versionCreated>
    <systemAttribute name="Capture-Date">2007-02-08T17:54:23.000Z</systemAttribute>
    <systemAttribute name="Height">160</systemAttribute>
    <systemAttribute name="Mime-Type">image/jpeg</systemAttribute>
    <systemAttribute name="Timeline-Date">2007-02-08T17:54:23.000Z</systemAttribute>
    <systemAttribute name="Width">160</systemAttribute>
    <changes>
      <change type="create">
        <detail name="path">/images/plane.jpg</detail>
      </change>
    </changes>
    <attributesChanged>true</attributesChanged>
    <deleted>false</deleted>
    <repository>desktop</repository>
    <uri>dv-file://6d20706ee27c447495d16642f37995c8:desktop/images/plane.jpg</uri>
    <commentCount>0</commentCount>
    <version>1</version>
    <checksum>fc1624ff53e9745bd64df7fdadbc7d864413d2743c0cc36d56a3199b806a5b1a</checksum>
    <extension>jpg</extension>
    <contentToken>AIJ2HrqDg1TiGaSpz2cNSz0f0dT8_GwDEvxY8UBNWOaEjQAQTQsi-mdkaTHy6YT5NRhD_L65l-
uLh_kR0FYN_mQ~</contentToken>
    <contentChanged>true</contentChanged>
  </file>
</files>
```

2

Folder APIs

These folder APIs can be used as alternatives to their equivalents in core Digital Vault.

2.1. dvx.user.folder.list (Deprecated)

Description

Deprecated. Use the core dv.user.folder.list API instead. Returns metadata for files and subfolders stored in multiple folders in one or more repositories. Note that the results are merged into a fake root folder element, of size 0, in the response document.

HTTP Method

GET

URI Structure

```
/user/{userUid}/folder
```

Table 2.1. URI Variables

Name	Description
`userUid`	The user's unique ID.

Request Headers

Calls to this API can include the following headers. Others headers may be required on a per project basis.

Table 2.2. Request Headers

Name	Requirement	Multiple	Type	Description
`Accept`	Optional	No	string	The following values are supported: • `application/vnd.newbay.dv-1.13+json` (default) • `application/vnd.newbay.dv-1.13+xml` • `application/vnd.newbay.dv-1.13+pbuf`

Request Parameters

This API call supports the following URI query parameters:

Table 2.3. URI Query Parameters

Name	Requirement	Multiple	Type	Description
`repositoryPath`	Mandatory	Yes	string	The repository path of the folder to list. For example, desktop:/images. A single request can include a maximum of 10 repositoryPath parameters.
`sort`	Optional	No	string	Controls the sort order in which the response is returned. Uses the syntax: sort={field}+(asc|desc)

Name	Requirement	Multiple	Type	Description
`start`	Optional	No	int	The start page in a paginated response. Defaults to 1. A link rel="next" element in the response document contain the URI to the page directly after the current page. A link rel="prev" element contains the URI to the page previous to the current page.
`count`	Optional	No	int	Specifies the number of items returned per page in a paginated response. Defaults to 20 and has a maximum value of 100, although both values are configurable on the server.
`filter`	Optional	No	string	Restricts the response document to contain only file or folder metadata. The value of this parameter must be either 'file' or 'folder' (case insensitive).

Response Codes

Successful responses return the following status codes:

- `200`

Failed operations return the following status codes:

- `400 401 403 404 406 415 500 502 503 504`

Supported Response Formats

On success, this API returns an object of type `folder`, as defined in the core Digital Vault API model.

- `application/vnd.newbay.dv-1.13+json`
- `application/vnd.newbay.dv-1.13+xml`
- `application/vnd.newbay.dv-1.13+pbuf`

Errors are returned in the following formats:

- `application/vnd.newbay.dv-1.13+json`
- `application/vnd.newbay.dv-1.13+xml`
- `application/vnd.newbay.dv-1.13+pbuf`

Sample Request and Response

Example 2.1. Request

```
GET /user/bb7b846b51e74c3fa09318d85363f80e/folder?repositoryPath=desktop%3A%2F&repositoryPath=Laptop%3A
%2F&filter=file
Authorization: NWB token=XXXXXXXXXXXXXXXX
Host: localhost:19006
User-Agent: Jakarta Commons-HttpClient/3.1
Accept: application/vnd.newbay.dv-1.14+xml
```

Example 2.2. Response

```
HTTP/1.1 200 OK
content-type: application/vnd.newbay.dv-1.14+xml;charset=UTF-8

<?xml version="1.0" encoding="UTF-8"?>
<folder totalCount="2" xmlns="http://dv.newbay.com/ns/1.0"
  xmlns:a="http://alternate.newbay.com/ns/1.0"
  xmlns:dvi="http://internal.dv.newbay.com/ns/1.0" xmlns:xspf="http://xspf.org/ns/0/">
  <link rel="self">http://localhost:19006/ext/user/bb7b846b51e74c3fa09318d85363f80e/folder?
repositoryPath=desktop%3A%2F&amp;repositoryPath=Laptop%3A%2F&amp;filter=file</link>
  <name>/</name>
  <parentPath>/</parentPath>
  <size>0</size>
  <versionCreated>1970-01-01T00:00:00.000Z</versionCreated>
  <changes>
    <change type="create">
      <detail name="path">/</detail>
    </change>
  </changes>
  <attributesChanged>false</attributesChanged>
  <deleted>false</deleted>
  <file>
    <link rel="file">http://localhost:19005/api/user/bb7b846b51e74c3fa09318d85363f80e/repository/
desktop/file?path=%2Fplane.jpg</link>
    <link rel="content">http://localhost:19005/api/user/bb7b846b51e74c3fa09318d85363f80e/repository/
desktop/file/content?path=%2Fplane.jpg</link>
    <name>plane.jpg</name>
    <parentPath>/</parentPath>
    <author>
      <phone>11111111001</phone>
      <email>dv_functional_test_alice001@synchronoss.com</email>
      <firstName>Alice</firstName>
      <lastName>Joyce</lastName>
    </author>
    <size>38592</size>
    <versionCreated>2015-03-02T10:53:16.955Z</versionCreated>
    <systemAttribute name="Capture-Date">2007-02-08T17:54:23.000Z</systemAttribute>
    <systemAttribute name="Height">160</systemAttribute>
    <systemAttribute name="Mime-Type">image/jpeg</systemAttribute>
    <systemAttribute name="Timeline-Date">2007-02-08T17:54:23.000Z</systemAttribute>
    <systemAttribute name="Width">160</systemAttribute>
    <changes>
      <change type="create">
        <detail name="path">/plane.jpg</detail>
      </change>
    </changes>
    <attributesChanged>true</attributesChanged>
    <deleted>false</deleted>
    <repository>desktop</repository>
    <uri>dv-file://bb7b846b51e74c3fa09318d85363f80e:desktop/plane.jpg</uri>
    <commentCount>0</commentCount>
    <version>1</version>
    <checksum>fc1624ff53e9745bd64df7fdadbc7d864413d2743c0cc36d56a3199b806a5b1a</checksum>
    <extension>jpg</extension>
    <fileAttribute name="ContentPermissions">SHARE</fileAttribute>
    <contentToken>AIJ2HrqDg1TiGaSpz2cNSz0f0dT8_GwDEvxY8UBNWOaEX-SdY1N6xiqYxMt3Bk2fnaB9FfwhmdQ4JW-
rjd8nwwk~</contentToken>
    <contentAccessible>true</contentAccessible>
    <contentChanged>true</contentChanged>
  </file>
  <file>
    <link rel="file">http://localhost:19005/api/user/bb7b846b51e74c3fa09318d85363f80e/repository/Laptop/
file?path=%2Fplane.jpg</link>
    <link rel="content">http://localhost:19005/api/user/bb7b846b51e74c3fa09318d85363f80e/repository/
Laptop/file/content?path=%2Fplane.jpg</link>
    <name>plane.jpg</name>
    <parentPath>/</parentPath>
    <author>
      <phone>11111111001</phone>
      <email>dv_functional_test_alice001@synchronoss.com</email>
      <firstName>Alice</firstName>
      <lastName>Joyce</lastName>
    </author>
    <size>38592</size>
    <versionCreated>2015-03-02T10:53:17.039Z</versionCreated>
    <systemAttribute name="Capture-Date">2007-02-08T17:54:23.000Z</systemAttribute>
    <systemAttribute name="Height">160</systemAttribute>
    <systemAttribute name="Mime-Type">image/jpeg</systemAttribute>
    <systemAttribute name="Timeline-Date">2007-02-08T17:54:23.000Z</systemAttribute>
    <systemAttribute name="Width">160</systemAttribute>
    <changes>
      <change type="create">
        <detail name="path">/plane.jpg</detail>
      </change>
    </changes>
    <attributesChanged>true</attributesChanged>
    <deleted>false</deleted>
    <repository>Laptop</repository>
    <uri>dv-file://bb7b846b51e74c3fa09318d85363f80e:Laptop/plane.jpg</uri>
    <commentCount>0</commentCount>
    <version>1</version>
    <checksum>fc1624ff53e9745bd64df7fdadbc7d864413d2743c0cc36d56a3199b806a5b1a</checksum>
    <extension>jpg</extension>
    <fileAttribute name="ContentPermissions">SHARE</fileAttribute>
    <contentToken>AIJ2HrqDg1TiGaSpz2cNSz0f0dT8_GwDEvxY8UBNWOaEX-SdY1N6xiqYxMt3Bk2fnaB9FfwhmdQ4JW-
rjd8nwwk~</contentToken>
    <contentAccessible>true</contentAccessible>
    <contentChanged>true</contentChanged>
  </file>
</folder>
```

2.2. dvx.user.repository.folder.create

Description

Creates a folder in a repository. Returns binary data, so can be used as an alternative to the core Digital Vault APIs, where you do not want XML or JSON in the response.

HTTP Method

POST

URI Structure

```
/user/{userUid}/repository/{repository}/folder
```

Table 2.4. URI Variables

Name	Description
`userUid`	The user's unique ID.
`repository`	The name of the user's repository.

Request Headers

Calls to this API can include the following headers. Others headers may be required on a per project basis.

Table 2.5. Request Headers

Name	Requirement	Multiple	Type	Description
`Content-Type`	Mandatory	No	string	The following values are supported: • `application/x-www-form-urlencoded`
`Accept`	Optional	No	string	The following values are supported: • `text/plain` (default) • `application/vnd.newbay.dv-1.13+json` • `application/vnd.newbay.dv-1.13+xml` • `application/vnd.newbay.dv-1.13+pbuf`

Request Parameters

This API call supports the following form parameters:

Table 2.6. Form Parameters

Name	Requirement	Multiple	Type	Description
`path`	Mandatory	No	string	The directory path to the folder's parent directory. Note that the parent path of a file system's root element is always /.
`name`	Mandatory	No	string	The name of the folder.

Response Codes

Successful responses return the following status codes:

- `201`

Failed operations return the following status codes:

- `400 401 403 404 406 415 500 502 503 504`

Supported Response Formats

On success, this API returns binary in the response body.

Errors are returned in the following formats:

- `application/vnd.newbay.dv-1.13+json`
- `application/vnd.newbay.dv-1.13+xml`
- `application/vnd.newbay.dv-1.13+pbuf`

Sample Request and Response

Example 2.3. Request

```
POST /user/6d89ab32e7544324836ecc0f2525a4c4/repository/desktop/folder
Authorization: NWB token=XXXXXXXXXXXXXXXX
Host: localhost:19006
Content-Length: 20
User-Agent: Jakarta Commons-HttpClient/3.1
Content-Type: application/x-www-form-urlencoded; charset=utf-8
Accept: application/vnd.newbay.dv-1.14+xml

name=bar&path=/foo
```

Example 2.4. Response

```

HTTP/1.1 201 Created
content-type: text/plain
content-length: 0

[binary stream - omitted]
```

3

Playlist APIs

Digital Vault Extensions allows users to create playlists, such as image slideshows or music playlists. Users can share their playlists using the Synchronoss Cloud Share component.

3.1. dvx.playlist.content.get

Description

Downloads a single file from a playlist.

HTTP Method

GET

URI Structure

```
/playlist/{playlistUid}/content
```

Table 3.1. URI Variables

Name	Description
`playlistUid`	The unique ID of the playlist.

Request Headers

Calls to this API can include the following headers. Others headers may be required on a per project basis.

Table 3.2. Request Headers

Name	Requirement	Multiple	Type	Description
`Accept`	Optional	No	string	The following values are supported: • `*/*` (default)

Request Parameters

This API call supports the following URI query parameters:

Table 3.3. URI Query Parameters

Name	Requirement	Multiple	Type	Description
`path`	Mandatory	No	string	The path to the file.
`alternateUid`	Optional	No	string	The unique ID of an alternate such as an album art image for a music playlist or an audio file for background music in a slideshow.

Response Codes

Successful responses return the following status codes:

- `200`

Failed operations return the following status codes:

- `400 401 403 404 406 415 500 502 503 504`

Supported Response Formats

On success, this API returns binary in the response body.

Errors are returned in the following formats:

- `application/vnd.newbay.dv-1.13+json`
- `application/vnd.newbay.dv-1.13+xml`
- `application/vnd.newbay.dv-1.13+pbuf`

Sample Request and Response

Example 3.1. Request

```
GET /playlist/f03136d280404823adfd6bec34926564/content?path=%2Ffoo.mp3
Authorization: NWB token=XXXXXXXXXXXXXXXX
Host: localhost:19006
User-Agent: Jakarta Commons-HttpClient/3.1
Accept: application/vnd.newbay.dv-1.14+xml
```

Example 3.2. Response

```
HTTP/1.1 200 OK
content-type: audio/mpeg
last-modified: Mon, 02 Mar 2015 10:53:11 GMT
etag: "05bcfe86afec4f8abb98433e49e11389c730e7bd46e8a0ad09d5a06eabe819b0"
content-length: 2855040
content-disposition: attachment; filename="foo.mp3"

[binary stream - omitted]
```

3.2. dvx.playlist.download

Description

Downloads a file containing a playlist file and all the media files contained in the playlist.

HTTP Method

GET

URI Structure

`/playlist/{playlistUid}/download`

Table 3.4. URI Variables

Name	Description
`playlistUid`	The unique ID of the playlist.

Request Headers

Calls to this API can include the following headers. Others headers may be required on a per project basis.

Table 3.5. Request Headers

Name	Requirement	Multiple	Type	Description
`Accept`	Optional	No	string	The following values are supported: • `application/zip` (default) • `application/vnd.newbay.dv-1.13+json` • `application/vnd.newbay.dv-1.13+xml` • `application/vnd.newbay.dv-1.13+pbuf`

Request Parameters

This API call supports the following URI query parameters:

Table 3.6. URI Query Parameters

Name	Requirement	Multiple	Type	Description
`format`	Mandatory	No	string	The format of the playlist file. Vault Extensions supports the M3U playlist format by default. Other formats can be defined at project level. The format should be specified using its media type name, for example 'zip+m3u'.
`name`	Optional	No	string	The name of the file to be created, defaults to 'playlist'. Filenames are case sensitive; can be up to 255 characters long; may not contain the following characters: * ? | ; : , ! @ # $ ( ) < > / \ " ' ` ~ { } [ ] = + & ^ <space> <tab>; cannot have a period at the start or the end; cannot be any of the following: com1, com2, ... com9; lpt1, lpt2, ... lpt9; con; nul; prn

Response Codes

Successful responses return the following status codes:

- `200`

Failed operations return the following status codes:

- `400 401 403 404 406 415 500 502 503 504`

Supported Response Formats

On success, this API returns binary in the response body.

Errors are returned in the following formats:

- `application/vnd.newbay.dv-1.13+json`
- `application/vnd.newbay.dv-1.13+xml`
- `application/vnd.newbay.dv-1.13+pbuf`

Sample Request and Response

Example 3.3. Request

```
GET /playlist/f03136d280404823adfd6bec34926564/download?format=zip%2Bm3u&name=
Authorization: NWB token=XXXXXXXXXXXXXXXX
Host: localhost:19006
User-Agent: Jakarta Commons-HttpClient/3.1
Accept: application/vnd.newbay.dv-1.14+xml
```

Example 3.4. Response

```
HTTP/1.1 200 OK
content-type: application/zip
content-disposition: attachment; filename="playlist.zip";filename*=UTF-8''"playlist.zip"

[binary stream - omitted]
```

3.3. dvx.playlist.list.get

Description

Retrieves a playlist's metadata and the metadata of all its entries. The metadata also includes any details set as system attributes, such as artist or album name.

HTTP Method

GET

URI Structure

`/playlist/{playlistUid}/list`

Table 3.7. URI Variables

Name	Description
`playlistUid`	The unique ID of the playlist.

Request Headers

Calls to this API can include the following headers. Others headers may be required on a per project basis.

Table 3.8. Request Headers

Name	Requirement	Multiple	Type	Description
`Accept`	Optional	No	string	The following values are supported: • `application/vnd.newbay.dv-1.13+json` (default) • `application/vnd.newbay.dv-1.13+xml` • `application/vnd.newbay.dv-1.13+pbuf`

Request Parameters

This API call supports the following URI query parameters:

Table 3.9. URI Query Parameters

Name	Requirement	Multiple	Type	Description
`sort`	Optional	No	string	Controls the sort order in which the response is returned. Use the syntax: sort={field}+(asc|desc)
`start`	Optional	No	int	The start page in a paginated response. Defaults to 1. A link rel="next" element in the response document contain the URI to the page directly after the current page. A link rel="prev" element contains the URI to the page previous to the current page.
`count`	Optional	No	int	Specifies the number of items returned per page in a paginated response. Defaults to 20 and has a maximum value of 100, although both values are configurable on the server.

Response Codes

Successful responses return the following status codes:

- `200`

Failed operations return the following status codes:

- `400 401 403 404 406 415 500 502 503 504`

Supported Response Formats

On success, this API returns an object of type **Playlist** in one of the following formats:

- `application/vnd.newbay.dv-1.13+json`
- `application/vnd.newbay.dv-1.13+xml`
- `application/vnd.newbay.dv-1.13+pbuf`

Errors are returned in the following formats:

- `application/vnd.newbay.dv-1.13+json`

- `application/vnd.newbay.dv-1.13+xml`
- `application/vnd.newbay.dv-1.13+pbuf`

Sample Request and Response

Example 3.5. Request

```
GET /user/e0962078ead144438c24a648a980a987/playlist/f03136d280404823adfd6bec34926564/list
Authorization: NWB token=XXXXXXXXXXXXXXXX
Host: localhost:19006
User-Agent: Jakarta Commons-HttpClient/3.1
Accept: application/vnd.newbay.dv-1.14+xml
```

Example 3.6. Response

```
HTTP/1.1 200 OK
content-type: application/vnd.newbay.dv-1.14+xml;charset=UTF-8

<?xml version="1.0" encoding="UTF-8"?>
<playlist pathCount="1" totalCount="1"
  xmlns="http://playlist.dv.newbay.com/ns/1.0" xmlns:a="http://alternate.newbay.com/ns/1.0">
  <link rel="self">http://localhost:19006/ext/playlist/f03136d280404823adfd6bec34926564/list</link>
  <link rel="playlist">http://localhost:19006/ext/playlist/f03136d280404823adfd6bec34926564/list</link>
  <uri>dv-playlist://f03136d280404823adfd6bec34926564</uri>
  <playlistElement>
    <link rel="content">http://localhost:19006/ext/playlist/f03136d280404823adfd6bec34926564/content?
path=%2Ffoo.mp3</link>
    <filename>foo.mp3</filename>
    <repositoryPath>desktop:/foo.mp3</repositoryPath>
    <mimeType>audio/mpeg</mimeType>
    <contentPermission>SHARE</contentPermission>
    <size>2855040</size>
    <versionCreated>2015-03-02T10:53:11.487Z</versionCreated>
    <pathId>6ycxer2a34</pathId>
    <fileAttribute name="ContentPermissions">SHARE</fileAttribute>
    <fileAttribute name="Duration">178000</fileAttribute>
    <contentToken>ALan-MxkSNBEfkwzDAENeGAuQBZA1pc-
najTes9wQXXL0bSKD0NkVX1GYYEApDmv3gXzCEPcKtCYCqDbDo87gAs~</contentToken>
    <checksum>05bcfe86afec4f8abb98433e49e11389c730e7bd46e8a0ad09d5a06eabe819b0</checksum>
    <contentAccessible>true</contentAccessible>
    <extension>mp3</extension>
    <title>URandom</title>
    <artist>Whistler</artist>
    <year>2009</year>
    <genre>Alternative</genre>
    <album>Ubuntu</album>
    <track>1</track>
    <viewUid type="root" viewType="playlist">f03136d280404823adfd6bec34926564</viewUid>
    <uri>dv-playlist-file://f03136d280404823adfd6bec34926564/foo.mp3</uri>
  </playlistElement>
</playlist>
```

3.4. dvx.playlist.uri.content.get

Description

Downloads a single file from a playlist using a using a URI, such as 'dv-playlist' or 'dv-playlist-share'.

HTTP Method

GET

URI Structure

```
/playlist/content
```

Request Headers

Calls to this API can include the following headers. Others headers may be required on a per project basis.

Table 3.10. Request Headers

Name	Requirement	Multiple	Type	Description
`Accept`	Optional	No	string	The following values are supported: • `*/*` (default)

Request Parameters

This API call supports the following URI query parameters:

Table 3.11. URI Query Parameters

Name	Requirement	Multiple	Type	Description
`uri`	Mandatory	No	string	The URI to the file to be downloaded.
`alternateUid`	Optional	No	string	The unique ID of an alternate such as an album art image for a music playlist or an audio file for background music in a slideshow.

Response Codes

Successful responses return the following status codes:

- `200`

Failed operations return the following status codes:

- `400 401 403 404 406 415 500 502 503 504`

Supported Response Formats

On success, this API returns binary in the response body.

Errors are returned in the following formats:

- `application/vnd.newbay.dv-1.13+json`
- `application/vnd.newbay.dv-1.13+xml`
- `application/vnd.newbay.dv-1.13+pbuf`

Sample Request and Response

Example 3.7. Request

```
GET /playlist/content?uri=dv-playlist-share%3A%2F%2FshareUid
%3A64741c308f7642cbbf90593e56669ae0%2Ffoo.mp3
Authorization: NWB token=XXXXXXXXXXXXXXXX
Host: localhost:19006
User-Agent: Jakarta Commons-HttpClient/3.1
Accept: application/vnd.newbay.dv-1.14+xml
```

Example 3.8. Response

```
HTTP/1.1 400 Bad Request
content-type: application/vnd.newbay.dv-1.14+xml;charset=UTF-8
x-error-code: 2201

<?xml version="1.0" encoding="UTF-8"?>
<errors xmlns="http://dv.newbay.com/ns/1.0"
  xmlns:a="http://alternate.newbay.com/ns/1.0"
  xmlns:dvi="http://internal.dv.newbay.com/ns/1.0" xmlns:xspf="http://xspf.org/ns/0/">
  <error code="2201">
    <message>Name is empty</message>
  </error>
</errors>
```

3.5. dvx.playlist.uri.download

Description

Uses a URI, such as 'dv-playlist' or 'dv-playlist-share', to download a file containing a playlist file and all the media files contained in the playlist.

HTTP Method

GET

URI Structure

`/playlist/download`

Request Headers

Calls to this API can include the following headers. Others headers may be required on a per project basis.

Table 3.12. Request Headers

Name	Requirement	Multiple	Type	Description
`Accept`	Optional	No	string	The following values are supported: • `application/zip` (default) • `application/vnd.newbay.dv-1.13+json` • `application/vnd.newbay.dv-1.13+xml` • `application/vnd.newbay.dv-1.13+pbuf`

Request Parameters

This API call supports the following URI query parameters:

Table 3.13. URI Query Parameters

Name	Requirement	Multiple	Type	Description
`uri`	Mandatory	No	string	The URI to the playlist.
`format`	Mandatory	No	string	The format of the playlist file. Vault Extensions supports the M3U playlist format by default. Other formats can be defined at project level. The format should be specified using its media type name, for example 'zip+m3u'.
`name`	Optional	No	string	The name of the file to be created, defaults to 'playlist'. Filenames are case sensitive; can be up to 255 characters long; may not contain the following characters: * ? | ; : , ! @ # $ ( ) < > / \ " ' ` ~ { } [ ] = + & ^ <space> <tab>; cannot have a period at the start or the end; cannot be any of the following: com1, com2, ... com9; lpt1, lpt2, ... lpt9; con; nul; prn

Response Codes

Successful responses return the following status codes:

- `200`

Failed operations return the following status codes:

- `400 401 403 404 406 415 500 502 503 504`

Supported Response Formats

On success, this API returns binary in the response body.

Errors are returned in the following formats:

- `application/vnd.newbay.dv-1.13+json`
- `application/vnd.newbay.dv-1.13+xml`
- `application/vnd.newbay.dv-1.13+pbuf`

Sample Request and Response

Example 3.9. Request

```
GET /playlist/download?uri=dv-playlist-share%3A%2F%2FshareUid
%3A64741c308f7642cbbf90593e56669ae0&format=zip%2Bm3u&name=
Authorization: NWB token=XXXXXXXXXXXXXXXX
Host: localhost:19006
User-Agent: Jakarta Commons-HttpClient/3.1
Accept: application/vnd.newbay.dv-1.14+xml
```

Example 3.10. Response

```
HTTP/1.1 200 OK
content-type: application/zip
content-disposition: attachment; filename="playlist.zip";filename*=UTF-8''"playlist.zip"

[binary stream - omitted]
```

3.6. dvx.playlist.uri.list.get

Description

Retrieves a playlist's metadata and the metadata of all its entries using a URI, such as 'dv-playlist' or 'dv-playlist-share'. The metadata also includes any details set as system attributes, such as artist or album name.

HTTP Method

GET

URI Structure

`/playlist/list`

Request Headers

Calls to this API can include the following headers. Others headers may be required on a per project basis.

Table 3.14. Request Headers

Name	Requirement	Multiple	Type	Description
`Accept`	Optional	No	string	The following values are supported: • `application/vnd.newbay.dv-1.13+json` (default) • `application/vnd.newbay.dv-1.13+xml` • `application/vnd.newbay.dv-1.13+pbuf`

Request Parameters

This API call supports the following URI query parameters:

Table 3.15. URI Query Parameters

Name	Requirement	Multiple	Type	Description
`uri`	Mandatory	No	string	The URI to the playlist.
`sort`	Optional	No	string	Controls the sort order in which the response is returned. Use the syntax: sort={field}+(asc|desc)
`start`	Optional	No	int	The start page in a paginated response. Defaults to 1. A link rel="next" element in the response document contain the URI to the page directly after the current page. A link rel="prev" element contains the URI to the page previous to the current page.
`count`	Optional	No	int	Specifies the number of items returned per page in a paginated response. Defaults to 20 and has a maximum value of 100, although both values are configurable on the server.

Response Codes

Successful responses return the following status codes:

- `200`

Failed operations return the following status codes:

- `400 401 403 404 406 415 500 502 503 504`

Supported Response Formats

On success, this API returns an object of type **Playlist** in one of the following formats:

- `application/vnd.newbay.dv-1.13+json`
- `application/vnd.newbay.dv-1.13+xml`
- `application/vnd.newbay.dv-1.13+pbuf`

Errors are returned in the following formats:

- `application/vnd.newbay.dv-1.13+json`
- `application/vnd.newbay.dv-1.13+xml`
- `application/vnd.newbay.dv-1.13+pbuf`

Sample Request and Response

Example 3.11. Request

```
GET /playlist/list?uri=dv-playlist-share%3A%2F%2FshareUid%3A64741c308f7642cbbf90593e56669ae0
Authorization: NWB token=XXXXXXXXXXXXXXXX
Host: localhost:19006
User-Agent: Jakarta Commons-HttpClient/3.1
```

```
Accept: application/vnd.newbay.dv-1.14+xml
```

Example 3.12. Response

```
HTTP/1.1 200 OK
content-type: application/vnd.newbay.dv-1.14+xml;charset=UTF-8

<?xml version="1.0" encoding="UTF-8"?>
<playlist pathCount="1" totalCount="1"
  xmlns="http://playlist.dv.newbay.com/ns/1.0" xmlns:a="http://alternate.newbay.com/ns/1.0">
  <link rel="self">http://localhost:19006/ext/playlist/64741c308f7642cbbf90593e56669ae0/list</link>
  <link rel="playlist">http://localhost:19006/ext/playlist/64741c308f7642cbbf90593e56669ae0/list</link>
  <uri>dv-playlist-share://shareUid:64741c308f7642cbbf90593e56669ae0</uri>
  <playlistElement>
    <link rel="content">http://localhost:19006/ext/playlist/content?uri=dv-playlist-share-file%3A%2F
%2FshareUid%3A64741c308f7642cbbf90593e56669ae0%2Ffoo.mp3</link>
    <filename>foo.mp3</filename>
    <repositoryPath>desktop:/foo.mp3</repositoryPath>
    <mimeType>audio/mpeg</mimeType>
    <contentPermission>SHARE</contentPermission>
    <size>2855040</size>
    <versionCreated>2015-03-02T10:53:17.460Z</versionCreated>
    <pathId>7ybggac9sl</pathId>
    <fileAttribute name="ContentPermissions">SHARE</fileAttribute>
    <fileAttribute name="Duration">178000</fileAttribute>
    <contentToken>ALan-MxkSNBEfkwzDAENeGAuQBZA1pc-najTes9wQXXLe5NXac3LoVd-frSIEVZkUdj-
YXp3GuVXJD5CpkXeCC8~</contentToken>
    <checksum>05bcfe86afec4f8abb98433e49e11389c730e7bd46e8a0ad09d5a06eabe819b0</checksum>
    <contentAccessible>true</contentAccessible>
    <extension>mp3</extension>
    <title>URandom</title>
    <artist>Whistler</artist>
    <year>2009</year>
    <genre>Alternative</genre>
    <album>Ubuntu</album>
    <track>1</track>
    <viewUid type="root" viewType="playlist">64741c308f7642cbbf90593e56669ae0</viewUid>
    <uri>dv-playlist-share-file://shareUid:64741c308f7642cbbf90593e56669ae0/foo.mp3</uri>
  </playlistElement>
</playlist>
```

3.7. dvx.provider.playlist.track.list

Description

Retrieves the tracks of a media provider playlist.

HTTP Method

GET

URI Structure

```
/provider/{provider}/playlist/{playlistUid}/track
```

Table 3.16. URI Variables

Name	Description
`provider`	The name of the media provider, for example, Napster.

Name	Description
`playlistUid`	The unique ID of the playlist.

Request Headers

Calls to this API can include the following headers. Others headers may be required on a per project basis.

Table 3.17. Request Headers

Name	Requirement	Multiple	Type	Description
`Accept`	Optional	No	string	The following values are supported: • `application/vnd.newbay.dv-1.13+json` (default) • `application/vnd.newbay.dv-1.13+xml` • `application/vnd.newbay.dv-1.13+pbuf`

Request Parameters

This API call supports the following URI query parameters:

Table 3.18. URI Query Parameters

Name	Requirement	Multiple	Type	Description
`start`	Optional	No	int	The start page in a paginated response. Defaults to 1. A link rel="next" element in the response document contain the URI to the page directly after the current page. A link rel="prev" element contains the URI to the page previous to the current page.
`count`	Optional	No	int	Specifies the number of items returned per page in a paginated response. Defaults to 20 and has a maximum value of 100, although both values are configurable on the server.
`sort`	Optional	No	string	Retrieves the tracks of a media provider playlist.

Response Codes

Successful responses return the following status codes:

- `200`

Failed operations return the following status codes:

- `400 401 403 404 406 415 500 502 503 504`

Supported Response Formats

On success, this API returns an object of type **Tracks** in one of the following formats:

- `application/vnd.newbay.dv-1.13+json`
- `application/vnd.newbay.dv-1.13+xml`
- `application/vnd.newbay.dv-1.13+pbuf`

Errors are returned in the following formats:

- `application/vnd.newbay.dv-1.13+json`
- `application/vnd.newbay.dv-1.13+xml`
- `application/vnd.newbay.dv-1.13+pbuf`

Sample Request and Response

Example 3.13. Request

```
GET /provider/napster/playlist/8be11a95f2434b81bbe0b8a4a2008ca2/track
Cookie: $Version=0; NWB=XXXXXXXXXXXXXXXX
Authorization: NWB token=XXXXXXXXXXXXXXXX
Host: localhost:19006
User-Agent: Jakarta Commons-HttpClient/3.1
Accept: application/vnd.newbay.dv-1.14+xml
```

Example 3.14. Response

```
HTTP/1.1 200 OK
content-type: application/vnd.newbay.dv-1.14+xml;charset=UTF-8

<?xml version="1.0" encoding="UTF-8"?>
<tracks totalCount="1"
  xmlns="http://playlist.dv.newbay.com/ns/1.0" xmlns:a="http://alternate.newbay.com/ns/1.0">
  <link rel="self">http://localhost:19006/ext/provider/napster/
playlist/8be11a95f2434b81bbe0b8a4a2008ca2/track</link>
  <track>
    <id>123</id>
    <title>test1.mp3</title>
    <artist>art1</artist>
    <album>alb1</album>
    <genre>Alternative</genre>
    <year>2009</year>
    <compilation>false</compilation>
    <track>1</track>
  </track>
</tracks>
```

3.8. dvx.user.playlist.create

Description

Creates a playlist.

HTTP Method

POST

URI Structure

```
/user/{userUid}/playlist
```

Table 3.19. URI Variables

Name	Description
`userUid`	The user's unique ID.

Request Headers

Calls to this API can include the following headers. Others headers may be required on a per project basis.

Table 3.20. Request Headers

Name	Requirement	Multiple	Type	Description
`Content-Type`	Mandatory	No	string	The following values are supported: • `application/vnd.newbay.dv-1.13+json` • `application/vnd.newbay.dv-1.13+xml` • `application/vnd.newbay.dv-1.13+pbuf`
`Accept`	Optional	No	string	The following values are supported: • `application/vnd.newbay.dv-1.13+json` (default) • `application/vnd.newbay.dv-1.13+xml` • `application/vnd.newbay.dv-1.13+pbuf`

Request Parameters

This API has no request parameters.

Request Body Format

Calls to this API must include an object of type **PlaylistDefinition** in the request body.

Response Codes

Successful responses return the following status codes:

- `201`

Failed operations return the following status codes:

- `400 401 403 404 406 415 500 502 503 504`

Supported Response Formats

On success, this API returns an object of type **PlaylistDefinition** in one of the following formats:

- `application/vnd.newbay.dv-1.13+json`
- `application/vnd.newbay.dv-1.13+xml`
- `application/vnd.newbay.dv-1.13+pbuf`

Errors are returned in the following formats:

- `application/vnd.newbay.dv-1.13+json`
- `application/vnd.newbay.dv-1.13+xml`
- `application/vnd.newbay.dv-1.13+pbuf`

Sample Request and Response

Example 3.15. Request

```
POST /user/e1e5d83561f44ead9608b5faa66131c7/playlist
Authorization: NWB token=XXXXXXXXXXXXXXXX
Host: localhost:19006
Content-Length: 286
User-Agent: Jakarta Commons-HttpClient/3.1
Content-Type: application/vnd.newbay.dv-1.14+xml
Accept: application/vnd.newbay.dv-1.14+xml

<?xml version="1.0" encoding="UTF-8"?>
<playlistDefinition
  xmlns="http://playlist.dv.newbay.com/ns/1.0" xmlns:a="http://alternate.newbay.com/ns/1.0">
  <name>playlistName</name>
  <type>music</type>
  <repositoryPath>desktop:/foo</repositoryPath>
  <repositoryPath>desktop:/bar</repositoryPath>
</playlistDefinition>
```

Example 3.16. Response

```
HTTP/1.1 201 Created
content-type: application/vnd.newbay.dv-1.14+xml;charset=UTF-8

<?xml version="1.0" encoding="UTF-8"?>
<playlistDefinition
  xmlns="http://playlist.dv.newbay.com/ns/1.0" xmlns:a="http://alternate.newbay.com/ns/1.0">
  <link rel="self">http://localhost:19006/ext/user/e1e5d83561f44ead9608b5faa66131c7/playlist/
a5d72e377c6d4518a1dd310f632d481a</link>
  <link rel="playlistDefinition">http://localhost:19006/ext/user/e1e5d83561f44ead9608b5faa66131c7/
playlist/a5d72e377c6d4518a1dd310f632d481a</link>
  <uid>a5d72e377c6d4518a1dd310f632d481a</uid>
  <name>playlistName</name>
  <type>music</type>
  <mimeTypeRegex>audio/.*</mimeTypeRegex>
  <repositoryPath pathId="nuriiuh6aa">desktop:/foo</repositoryPath>
  <repositoryPath pathId="ta5n66wact">desktop:/bar</repositoryPath>
  <creationDate>2015-03-02T10:53:13.746Z</creationDate>
  <lastModifiedDate>2015-03-02T10:53:13.746Z</lastModifiedDate>
  <uri>dv-playlist://a5d72e377c6d4518a1dd310f632d481a</uri>
  <validPathCount>0</validPathCount>
</playlistDefinition>
```

3.9. dvx.user.playlist.delete

Description

Deletes a playlist.

HTTP Method

DELETE

URI Structure

```
/user/{userUid}/playlist/{playlistUid}
```

Table 3.21. URI Variables

Name	Description
`userUid`	The user's unique ID.
`playlistUid`	The unique ID of the playlist.

Request Headers

Calls to this API can include the following headers. Others headers may be required on a per project basis.

Table 3.22. Request Headers

Name	Requirement	Multiple	Type	Description
`Accept`	Optional	No	string	The following values are supported: • `application/vnd.newbay.dv-1.13+json` (default) • `application/vnd.newbay.dv-1.13+xml` • `application/vnd.newbay.dv-1.13+pbuf`

Request Parameters

This API has no request parameters.

Response Codes

Successful responses return the following status codes:

- `200`

Failed operations return the following status codes:

- `400 401 403 404 406 415 500 502 503 504`

Supported Response Formats

On success, this API returns an empty response document.

Errors are returned in the following formats:

- `application/vnd.newbay.dv-1.13+json`
- `application/vnd.newbay.dv-1.13+xml`
- `application/vnd.newbay.dv-1.13+pbuf`

Sample Request and Response

Example 3.17. Request

```
DELETE /user/e1e5d83561f44ead9608b5faa66131c7/playlist/a5d72e377c6d4518a1dd310f632d481a
Authorization: NWB token=XXXXXXXXXXXXXXXX
Host: localhost:19006
User-Agent: Jakarta Commons-HttpClient/3.1
Accept: application/vnd.newbay.dv-1.14+xml
```

Example 3.18. Response

```
HTTP/1.1 200 OK
content-type: application/vnd.newbay.dv-1.14+xml;charset=UTF-8
content-length: 0
```

3.10. dvx.user.playlist.get

Description

Returns a playlist definition.

HTTP Method

GET

URI Structure

`/user/{userUid}/playlist/{playlistUid}`

Table 3.23. URI Variables

Name	Description
`userUid`	The user's unique ID.
`playlistUid`	The unique ID of the playlist.

Request Headers

Calls to this API can include the following headers. Others headers may be required on a per project basis.

Table 3.24. Request Headers

Name	Requirement	Multiple	Type	Description
`Accept`	Optional	No	string	The following values are supported: • `application/vnd.newbay.dv-1.13+json` (default) • `application/vnd.newbay.dv-1.13+xml` • `application/vnd.newbay.dv-1.13+pbuf`

Request Parameters

This API has no request parameters.

Response Codes

Successful responses return the following status codes:

- `200`

Failed operations return the following status codes:

- `400 401 403 404 406 415 500 502 503 504`

Supported Response Formats

On success, this API returns an object of type **PlaylistDefinition** in one of the following formats:

- `application/vnd.newbay.dv-1.13+json`
- `application/vnd.newbay.dv-1.13+xml`
- `application/vnd.newbay.dv-1.13+pbuf`

Errors are returned in the following formats:

- `application/vnd.newbay.dv-1.13+json`
- `application/vnd.newbay.dv-1.13+xml`
- `application/vnd.newbay.dv-1.13+pbuf`

Sample Request and Response

Example 3.19. Request

```
GET /user/e1e5d83561f44ead9608b5faa66131c7/playlist/a5d72e377c6d4518a1dd310f632d481a
Authorization: NWB token=XXXXXXXXXXXXXXXX
Host: localhost:19006
User-Agent: Jakarta Commons-HttpClient/3.1
Accept: application/vnd.newbay.dv-1.14+xml
```

Example 3.20. Response

```
HTTP/1.1 200 OK
content-type: application/vnd.newbay.dv-1.14+xml;charset=UTF-8

<?xml version="1.0" encoding="UTF-8"?>
<playlistDefinition
  xmlns="http://playlist.dv.newbay.com/ns/1.0" xmlns:a="http://alternate.newbay.com/ns/1.0">
  <link rel="self">http://localhost:19006/ext/user/e1e5d83561f44ead9608b5faa66131c7/playlist/
a5d72e377c6d4518a1dd310f632d481a</link>
  <link rel="playlistDefinition">http://localhost:19006/ext/user/e1e5d83561f44ead9608b5faa66131c7/
playlist/a5d72e377c6d4518a1dd310f632d481a</link>
  <uid>a5d72e377c6d4518a1dd310f632d481a</uid>
  <name>playlistName</name>
  <type>music</type>
  <mimeTypeRegex>audio/.*</mimeTypeRegex>
  <repositoryPath pathId="nuriiuh6aa">desktop:/foo</repositoryPath>
  <repositoryPath pathId="ta5n66wact">desktop:/bar</repositoryPath>
  <creationDate>2015-03-02T10:53:13.746Z</creationDate>
  <lastModifiedDate>2015-03-02T10:53:13.746Z</lastModifiedDate>
  <uri>dv-playlist://a5d72e377c6d4518a1dd310f632d481a</uri>
  <validPathCount>0</validPathCount>
</playlistDefinition>
```

3.11. dvx.user.playlist.list

Description

Returns a list of all playlists owned by the user.

HTTP Method

GET

URI Structure

`/user/{userUid}/playlist`

Table 3.25. URI Variables

Name	Description
`userUid`	The user's unique ID.

Request Headers

Calls to this API can include the following headers. Others headers may be required on a per project basis.

Table 3.26. Request Headers

Name	Requirement	Multiple	Type	Description
`Accept`	Optional	No	string	The following values are supported: • `application/vnd.newbay.dv-1.13+json` (default) • `application/vnd.newbay.dv-1.13+xml` • `application/vnd.newbay.dv-1.13+pbuf`

Request Parameters

This API call supports the following URI query parameters:

Table 3.27. URI Query Parameters

Name	Requirement	Multiple	Type	Description
`type`	Optional	Yes	string	Allows the client to request playlists of a certain type, such as 'audio', 'image', 'video' or 'image-video'.
`start`	Optional	No	int	The start page in a paginated response. Defaults to 1. A link rel="next" element in the response document contain the URI to the page directly after the current page. A link rel="prev" element contains the URI to the page previous to the current page.
`count`	Optional	No	int	Specifies the number of items returned per page in a paginated response. Defaults to 20 and has a maximum value of 100, although both values are configurable on the server.
`hideExpired`	Optional	No	boolean	If 'true', expired playlists are omitted from the response document.
`hideNotShared`	Optional	No	boolean	If 'true', only shared playlists are shown in the response document.
`sort`	Optional	No	string	Controls the sort order in which the response is returned. Uses the syntax: sort={field}+(asc|desc), where {field} can be 'name' or 'creationDate'.

Response Codes

Successful responses return the following status codes:

- `200`

Failed operations return the following status codes:

- `400 401 403 404 406 415 500 502 503 504`

Supported Response Formats

On success, this API returns an object of type **PlaylistDefinitions** in one of the following formats:

- `application/vnd.newbay.dv-1.13+json`
- `application/vnd.newbay.dv-1.13+xml`
- `application/vnd.newbay.dv-1.13+pbuf`

Errors are returned in the following formats:

- `application/vnd.newbay.dv-1.13+json`
- `application/vnd.newbay.dv-1.13+xml`
- `application/vnd.newbay.dv-1.13+pbuf`

Sample Request and Response

Example 3.21. Request

```
GET /user/e1e5d83561f44ead9608b5faa66131c7/playlist
Authorization: NWB token=XXXXXXXXXXXXXXXX
Host: localhost:19006
User-Agent: Jakarta Commons-HttpClient/3.1
Accept: application/vnd.newbay.dv-1.14+xml
```

Example 3.22. Response

```
HTTP/1.1 200 OK
content-type: application/vnd.newbay.dv-1.14+xml;charset=UTF-8

<?xml version="1.0" encoding="UTF-8"?>
<playlistDefinitions totalCount="1"
  xmlns="http://playlist.dv.newbay.com/ns/1.0" xmlns:a="http://alternate.newbay.com/ns/1.0">
  <link rel="self">http://localhost:19006/ext/user/e1e5d83561f44ead9608b5faa66131c7/playlist</link>
  <link rel="playlistDefinitions">http://localhost:19006/ext/user/e1e5d83561f44ead9608b5faa66131c7/
playlist</link>
  <playlistDefinition>
    <link rel="playlistDefinition">http://localhost:19006/ext/user/e1e5d83561f44ead9608b5faa66131c7/
playlist/a5d72e377c6d4518a1dd310f632d481a</link>
    <uid>a5d72e377c6d4518a1dd310f632d481a</uid>
    <name>playlistName</name>
    <type>music</type>
    <mimeTypeRegex>audio/.*</mimeTypeRegex>
    <repositoryPath pathId="nuriiuh6aa">desktop:/foo</repositoryPath>
    <repositoryPath pathId="ta5n66wact">desktop:/bar</repositoryPath>
    <creationDate>2015-03-02T10:53:13.746Z</creationDate>
    <lastModifiedDate>2015-03-02T10:53:13.746Z</lastModifiedDate>
    <uri>dv-playlist://a5d72e377c6d4518a1dd310f632d481a</uri>
    <validPathCount>0</validPathCount>
  </playlistDefinition>
</playlistDefinitions>
```

3.12. dvx.user.playlist.metadata.update

Description

Allows a client to update a playlist by sending a new 'playlistDefinition' model in a POST request but without having to include any 'repositoryPath' elements in the model.

HTTP Method

POST

URI Structure

`/user/{userUid}/playlist/{playlistUid}/metadata`

Table 3.28. URI Variables

Name	Description
`userUid`	The user's unique ID.

Name	Description
`playlistUid`	The unique ID of the playlist.

Request Headers

Calls to this API can include the following headers. Others headers may be required on a per project basis.

Table 3.29. Request Headers

Name	Requirement	Multiple	Type	Description
`Content-Type`	Mandatory	No	string	The following values are supported: • `application/vnd.newbay.dv-1.13+json` • `application/vnd.newbay.dv-1.13+xml` • `application/vnd.newbay.dv-1.13+pbuf`
`Accept`	Optional	No	string	The following values are supported: • `application/vnd.newbay.dv-1.13+json` (default) • `application/vnd.newbay.dv-1.13+xml` • `application/vnd.newbay.dv-1.13+pbuf`

Request Parameters

This API has no request parameters.

Request Body Format

Calls to this API must include an object of type **PlaylistDefinition** in the request body.

Response Codes

Successful responses return the following status codes:

- `200 201`

Failed operations return the following status codes:

- `400 401 403 404 406 415 500 502 503 504`

Supported Response Formats

On success, this API returns an object of type **PlaylistDefinition** in one of the following formats:

- `application/vnd.newbay.dv-1.13+json`
- `application/vnd.newbay.dv-1.13+xml`

- `application/vnd.newbay.dv-1.13+pbuf`

Errors are returned in the following formats:

- `application/vnd.newbay.dv-1.13+json`
- `application/vnd.newbay.dv-1.13+xml`
- `application/vnd.newbay.dv-1.13+pbuf`

Sample Request and Response

Example 3.23. Request

```
POST /user/e1e5d83561f44ead9608b5faa66131c7/playlist/a5d72e377c6d4518a1dd310f632d481a/metadata
Authorization: NWB token=XXXXXXXXXXXXXXXX
Host: localhost:19006
Content-Length: 206
User-Agent: Jakarta Commons-HttpClient/3.1
Content-Type: application/vnd.newbay.dv-1.14+xml
Accept: application/vnd.newbay.dv-1.14+xml

<?xml version="1.0" encoding="UTF-8"?>
<playlistDefinition
  xmlns="http://playlist.dv.newbay.com/ns/1.0" xmlns:a="http://alternate.newbay.com/ns/1.0">
  <name>anotherNewPlaylistName</name>
  <type>music</type>
</playlistDefinition>
```

Example 3.24. Response

```
HTTP/1.1 200 OK
content-type: application/vnd.newbay.dv-1.14+xml;charset=UTF-8

<?xml version="1.0" encoding="UTF-8"?>
<playlistDefinition
  xmlns="http://playlist.dv.newbay.com/ns/1.0" xmlns:a="http://alternate.newbay.com/ns/1.0">
  <link rel="playlistDefinition">http://localhost:19006/ext/user/e1e5d83561f44ead9608b5faa66131c7/
playlist/a5d72e377c6d4518a1dd310f632d481a</link>
  <uid>a5d72e377c6d4518a1dd310f632d481a</uid>
  <name>anotherNewPlaylistName</name>
  <type>music</type>
  <mimeTypeRegex>audio/.*</mimeTypeRegex>
  <repositoryPath pathId="mhhie3yg8l">desktop:/foo</repositoryPath>
  <repositoryPath pathId="g1cquds7ya">desktop:/bar</repositoryPath>
  <repositoryPath pathId="gz9eueqkg8">desktop:/foobar</repositoryPath>
  <creationDate>2015-03-02T10:53:13.746Z</creationDate>
  <lastModifiedDate>2015-03-02T10:53:14.126Z</lastModifiedDate>
  <uri>dv-playlist://a5d72e377c6d4518a1dd310f632d481a</uri>
  <validPathCount>0</validPathCount>
</playlistDefinition>
```

3.13. dvx.user.playlist.paths.update

Description

Allows a client to add or remove files from a playlist.

HTTP Method

POST

URI Structure

```
/user/{userUid}/playlist/{playlistUid}/paths
```

Table 3.30. URI Variables

Name	Description
userUid	The user's unique ID.
playlistUid	The unique ID of the playlist.

Request Headers

Calls to this API can include the following headers. Others headers may be required on a per project basis.

Table 3.31. Request Headers

Name	Requirement	Multiple	Type	Description
Content-Type	Mandatory	No	string	The following values are supported: • application/x-www-form-urlencoded
Accept	Optional	No	string	The following values are supported: • application/vnd.newbay.dv-1.13+json (default) • application/vnd.newbay.dv-1.13+xml • application/vnd.newbay.dv-1.13+pbuf

Request Parameters

This API call supports the following form parameters:

Table 3.32. Form Parameters

Name	Requirement	Multiple	Type	Description
add	Optional	Yes	string	The path to the file to be added to the playlist, for example, 'desktop://images/holidaysnaps/ourhotel.jpg'.
remove	Optional	Yes	string	The file to be removed from the playlist, as specified by a path ID, for example, 'xl92qu06e1'.

Response Codes

Successful responses return the following status codes:

- `200 201`

Failed operations return the following status codes:

- `400 401 403 404 406 415 500 502 503 504`

Supported Response Formats

On success, this API returns an object of type **RepositoryPaths** in one of the following formats:

- `application/vnd.newbay.dv-1.13+json`
- `application/vnd.newbay.dv-1.13+xml`
- `application/vnd.newbay.dv-1.13+pbuf`

Errors are returned in the following formats:

- `application/vnd.newbay.dv-1.13+json`
- `application/vnd.newbay.dv-1.13+xml`
- `application/vnd.newbay.dv-1.13+pbuf`

Sample Request and Response

Example 3.25. Request

```
POST /user/e1e5d83561f44ead9608b5faa66131c7/playlist/a5d72e377c6d4518a1dd310f632d481a/paths
Authorization: NWB token=XXXXXXXXXXXXXXXX
Host: localhost:19006
Content-Length: 38
User-Agent: Jakarta Commons-HttpClient/3.1
Content-Type: application/x-www-form-urlencoded; charset=utf-8
Accept: application/vnd.newbay.dv-1.14+xml

remove=mhhie3yg8l&add=desktop:/new
```

Example 3.26. Response

```
HTTP/1.1 200 OK
content-type: application/vnd.newbay.dv-1.14+xml;charset=UTF-8

<?xml version="1.0" encoding="UTF-8"?>
<repositoryPaths
  xmlns="http://playlist.dv.newbay.com/ns/1.0" xmlns:a="http://alternate.newbay.com/ns/1.0">
  <repositoryPath pathId="shfo6renhp">desktop:/new</repositoryPath>
</repositoryPaths>
```

3.14. dvx.user.playlist.reorder

Description

Reorder items in a playlist.

HTTP Method

POST

URI Structure

```
/user/{userUid}/playlist/{playlistUid}/reorder
```

Table 3.33. URI Variables

Name	Description
`userUid`	The user's unique ID.
`playlistUid`	The unique ID of the playlist.

Request Headers

Calls to this API can include the following headers. Others headers may be required on a per project basis.

Table 3.34. Request Headers

Name	Requirement	Multiple	Type	Description
`Content-Type`	Mandatory	No	string	The following values are supported: • `application/x-www-form-urlencoded`
`Accept`	Optional	No	string	The following values are supported: • `application/vnd.newbay.dv-1.13+json` (default) • `application/vnd.newbay.dv-1.13+xml` • `application/vnd.newbay.dv-1.13+pbuf`

Request Parameters

This API call supports the following form parameters:

Table 3.35. Form Parameters

Name	Requirement	Multiple	Type	Description
`fromPathId`	Mandatory	No	string	The current position of the playlist item.
`toPathId`	Mandatory	No	string	The new position for the playlist item.

Response Codes

Successful responses return the following status codes:

- `200`

Failed operations return the following status codes:

- `400 401 403 404 406 415 500 502 503 504`

Supported Response Formats

On success, this API returns an object of type **PlaylistDefinition** in one of the following formats:

- `application/vnd.newbay.dv-1.13+json`

- `application/vnd.newbay.dv-1.13+xml`
- `application/vnd.newbay.dv-1.13+pbuf`

Errors are returned in the following formats:

- `application/vnd.newbay.dv-1.13+json`
- `application/vnd.newbay.dv-1.13+xml`
- `application/vnd.newbay.dv-1.13+pbuf`

Sample Request and Response

Example 3.27. Request

```
POST /user/e1e5d83561f44ead9608b5faa66131c7/playlist/a5d72e377c6d4518a1dd310f632d481a/reorder
Authorization: NWB token=XXXXXXXXXXXXXXXX
Host: localhost:19006
Content-Length: 41
User-Agent: Jakarta Commons-HttpClient/3.1
Content-Type: application/x-www-form-urlencoded; charset=utf-8
Accept: application/vnd.newbay.dv-1.14+xml

toPathId=ta5n66wact&fromPathId=nuriiuh6aa
```

Example 3.28. Response

```
HTTP/1.1 200 OK
content-type: application/vnd.newbay.dv-1.14+xml;charset=UTF-8

<?xml version="1.0" encoding="UTF-8"?>
<playlistDefinition
  xmlns="http://playlist.dv.newbay.com/ns/1.0" xmlns:a="http://alternate.newbay.com/ns/1.0">
  <link rel="playlistDefinition">http://localhost:19006/ext/user/e1e5d83561f44ead9608b5faa66131c7/
playlist/a5d72e377c6d4518a1dd310f632d481a</link>
  <uid>a5d72e377c6d4518a1dd310f632d481a</uid>
  <name>playlistName</name>
  <type>music</type>
  <mimeTypeRegex>audio/.*</mimeTypeRegex>
  <repositoryPath pathId="ta5n66wact">desktop:/bar</repositoryPath>
  <repositoryPath pathId="nuriiuh6aa">desktop:/foo</repositoryPath>
  <creationDate>2015-03-02T10:53:13.746Z</creationDate>
  <lastModifiedDate>2015-03-02T10:53:13.967Z</lastModifiedDate>
  <uri>dv-playlist://a5d72e377c6d4518a1dd310f632d481a</uri>
  <validPathCount>0</validPathCount>
</playlistDefinition>
```

3.15. dvx.user.playlist.update

Description

Allows a client to update a playlist by sending a new 'playlistDefinition' model in a POST request.

HTTP Method

POST

URI Structure

```
/user/{userUid}/playlist/{playlistUid}
```

Table 3.36. URI Variables

Name	Description
`userUid`	The user's unique ID.
`playlistUid`	The unique ID of the playlist.

Request Headers

Calls to this API can include the following headers. Others headers may be required on a per project basis.

Table 3.37. Request Headers

Name	Requirement	Multiple	Type	Description
`Content-Type`	Mandatory	No	string	The following values are supported: • `application/vnd.newbay.dv-1.13+json` • `application/vnd.newbay.dv-1.13+xml` • `application/vnd.newbay.dv-1.13+pbuf`
`Accept`	Optional	No	string	The following values are supported: • `application/vnd.newbay.dv-1.13+json` (default) • `application/vnd.newbay.dv-1.13+xml` • `application/vnd.newbay.dv-1.13+pbuf`

Request Parameters

This API has no request parameters.

Request Body Format

Calls to this API must include an object of type **PlaylistDefinition** in the request body.

Response Codes

Successful responses return the following status codes:

- `200`

Failed operations return the following status codes:

- `400 401 403 404 406 415 500 502 503 504`

Supported Response Formats

On success, this API returns an object of type **PlaylistDefinition** in one of the following formats:

- `application/vnd.newbay.dv-1.13+json`
- `application/vnd.newbay.dv-1.13+xml`
- `application/vnd.newbay.dv-1.13+pbuf`

Errors are returned in the following formats:

- `application/vnd.newbay.dv-1.13+json`
- `application/vnd.newbay.dv-1.13+xml`
- `application/vnd.newbay.dv-1.13+pbuf`

Sample Request and Response

Example 3.29. Request

```
POST /user/e1e5d83561f44ead9608b5faa66131c7/playlist/a5d72e377c6d4518a1dd310f632d481a
Authorization: NWB token=XXXXXXXXXXXXXXXX
Host: localhost:19006
Content-Length: 337
User-Agent: Jakarta Commons-HttpClient/3.1
Content-Type: application/vnd.newbay.dv-1.14+xml
Accept: application/vnd.newbay.dv-1.14+xml

<?xml version="1.0" encoding="UTF-8"?>
<playlistDefinition
  xmlns="http://playlist.dv.newbay.com/ns/1.0" xmlns:a="http://alternate.newbay.com/ns/1.0">
  <name>newPlaylistName</name>
  <type>music</type>
  <repositoryPath>desktop:/foo</repositoryPath>
  <repositoryPath>desktop:/bar</repositoryPath>
  <repositoryPath>desktop:/foobar</repositoryPath>
</playlistDefinition>
```

Example 3.30. Response

```
HTTP/1.1 200 OK
content-type: application/vnd.newbay.dv-1.14+xml;charset=UTF-8

<?xml version="1.0" encoding="UTF-8"?>
<playlistDefinition
  xmlns="http://playlist.dv.newbay.com/ns/1.0" xmlns:a="http://alternate.newbay.com/ns/1.0">
  <link rel="playlistDefinition">http://localhost:19006/ext/user/e1e5d83561f44ead9608b5faa66131c7/
playlist/a5d72e377c6d4518a1dd310f632d481a</link>
  <uid>a5d72e377c6d4518a1dd310f632d481a</uid>
  <name>newPlaylistName</name>
  <type>music</type>
  <mimeTypeRegex>audio/.*</mimeTypeRegex>
  <repositoryPath pathId="mhhie3yg8l">desktop:/foo</repositoryPath>
  <repositoryPath pathId="g1cquds7ya">desktop:/bar</repositoryPath>
  <repositoryPath pathId="gz9eueqkg8">desktop:/foobar</repositoryPath>
  <creationDate>2015-03-02T10:53:13.746Z</creationDate>
  <lastModifiedDate>2015-03-02T10:53:14.126Z</lastModifiedDate>
  <uri>dv-playlist://a5d72e377c6d4518a1dd310f632d481a</uri>
  <validPathCount>0</validPathCount>
</playlistDefinition>
```

3.16. dvx.user.provider.playlist.list

Description

Returns a list of all playlists owned by the user that contain at least one file found within the context of the media provider.

HTTP Method

GET

URI Structure

```
/user/{userUid}/provider/{provider}/playlist
```

Table 3.38. URI Variables

Name	Description
`userUid`	The user's unique ID.
`provider`	The name of the media provider, for example, Napster.

Request Headers

Calls to this API can include the following headers. Others headers may be required on a per project basis.

Table 3.39. Request Headers

Name	Requirement	Multiple	Type	Description
`Accept`	Optional	No	string	The following values are supported: • `application/vnd.newbay.dv-1.13+json` (default) • `application/vnd.newbay.dv-1.13+xml` • `application/vnd.newbay.dv-1.13+pbuf`

Request Parameters

This API call supports the following URI query parameters:

Table 3.40. URI Query Parameters

Name	Requirement	Multiple	Type	Description
`start`	Optional	No	int	The start page in a paginated response. Defaults to 1. A link rel="next" element in the response document contain the URI to the page directly after the current page. A link rel="prev" element contains the URI to the page previous to the current page.

Name	Requirement	Multiple	Type	Description
`count`	Optional	No	int	Specifies the number of items returned per page in a paginated response. Defaults to 20 and has a maximum value of 100, although both values are configurable on the server.
`sort`	Optional	No	string	Returns a list of all playlists owned by the user that contain at least one file found within the context of the media provider.

Response Codes

Successful responses return the following status codes:

- `200`

Failed operations return the following status codes:

- `400 401 403 404 406 415 500 502 503 504`

Supported Response Formats

On success, this API returns an object of type **TrackPlaylistDefinitions** in one of the following formats:

- `application/vnd.newbay.dv-1.13+json`
- `application/vnd.newbay.dv-1.13+xml`
- `application/vnd.newbay.dv-1.13+pbuf`

Errors are returned in the following formats:

- `application/vnd.newbay.dv-1.13+json`
- `application/vnd.newbay.dv-1.13+xml`
- `application/vnd.newbay.dv-1.13+pbuf`

Sample Request and Response

Example 3.31. Request

```
GET /user/e6e7817d84d346e18d2a30c7a2045dcc/provider/napster/playlist
Authorization: NWB token=XXXXXXXXXXXXXXXX
Host: localhost:19006
User-Agent: Jakarta Commons-HttpClient/3.1
Accept: application/vnd.newbay.dv-1.14+xml
```

Example 3.32. Response

```
HTTP/1.1 200 OK
content-type: application/vnd.newbay.dv-1.14+xml;charset=UTF-8

<?xml version="1.0" encoding="UTF-8"?>
<trackPlaylistDefinitions totalCount="5"
```

```
  xmlns="http://playlist.dv.newbay.com/ns/1.0" xmlns:a="http://alternate.newbay.com/ns/1.0">
  <link rel="self">http://localhost:19006/ext/user/e6e7817d84d346e18d2a30c7a2045dcc/provider/napster/
playlist</link>
  <trackPlaylistDefinition>
    <uid>1244976d973c49fc8656dbdf9773489e</uid>
    <name>playlistName1</name>
    <trackCount>1</trackCount>
  </trackPlaylistDefinition>
  <trackPlaylistDefinition>
    <uid>34f22f057a024c4dab7fa1f91ce5e81d</uid>
    <name>playlistName2</name>
    <trackCount>1</trackCount>
  </trackPlaylistDefinition>
  <trackPlaylistDefinition>
    <uid>15550effb75e4cef9ee54a446d768158</uid>
    <name>playlistName3</name>
    <trackCount>1</trackCount>
  </trackPlaylistDefinition>
  <trackPlaylistDefinition>
    <uid>905960f760a14ce9b28c44005ed4ab06</uid>
    <name>playlistName4</name>
    <trackCount>1</trackCount>
  </trackPlaylistDefinition>
  <trackPlaylistDefinition>
    <uid>3bbe777774db406bb897a17e24ef87d2</uid>
    <name>playlistName5</name>
    <trackCount>1</trackCount>
  </trackPlaylistDefinition>
</trackPlaylistDefinitions>
```

A

Digital Vault Extensions Playlist Data Model

This appendix describes the document model used to represent content sent between client applications and the Digital Vault Extensions API server.

A.1. Model Overview

The Digital Vault Extensions API server accepts and returns data in a format determined by an HTTP `Accept` header contained in the HTTP request. The format can be XML, JSON or Protocol Buffers (Protobuf).

The XML, JSON and Protobuf are derived from models defined by the Digital Vault schema, written in the W3C XML Schema language. The schema consists of complex types that consist of element children and attributes

A.1.1. Root Elements

The root element of all the supported response documents are listed in the "Root Types" table. The **Name** column lists the name of the root element. The **Type** column lists the type associated with the root element. The **Description** column contains a brief description of the root element. The **Since** column contains the version of the Digital Vault model the root element was introduced.

A.1.2. Types

All the supported complex types are listed in the "Types" table. The **Name** column lists the type name. The **Description** column contains a brief description of the type. The **Since** column contains the version of the Digital Vault API model the type was introduced.

A.1.3. Type Descriptions

Each complex type is described individually in its own section. The **Name** column lists each child element and attribute contained within the type. The **Type** column describes the type (simple or complex) associated with each element or attribute. The **Description** column contains a brief description of what type of content the element or attribute contains. The **Cardinality** column describes the number of possible occurrences for the element or attribute. A value of `0..1` means zero OR one occurrence can occur. A value of `0..*` means zero OR any number can occur (i.e. unbounded). A value of `1..1` means one occurrence is mandatory. The **Since** column contains the version of the Digital Vault API model in which the element or attribute was introduced.

A.2. Root Types

Table A.1. Root Types

Name	Type	Description	Since
playlist	**Playlist**		1.0
playlistDefinition	**PlaylistDefinition**		1.0
playlistDefinitions	**PlaylistDefinitions**		1.0
playlistMembership	**PlaylistMembership**	Deprecated. Playlists no longer support the concept of 'membership'.	1.0
playlistMemberships	**PlaylistMemberships**	Deprecated. Playlists no longer support the concept of 'membership'.	1.0
repositoryPaths	**RepositoryPaths**		1.10

Name	Type	Description	Since
taggedContact	**TaggedContact**		1.14
taggedContacts	**TaggedContacts**		1.14
track	**Track**		1.13
trackPlaylistDefinition	**TrackPlaylistDefinition**		1.13
trackPlaylistDefinitions	**TrackPlaylistDefinitions**		1.13
tracks	**Tracks**		1.13

A.3. Types

Table A.2. Types

Type	Description	Since
Attribute	Attribute	1.0
AttributedRepositoryPath	AttributedRepositoryPath	1.0
Checksum	Checksum	1.0
InvalidPlaylistElement	InvalidPlaylistElement	1.0
Link	Link	1.0
PagedList	PagedList	1.0
Playlist	Playlist	1.0
PlaylistDefinition	PlaylistDefinition	1.0
PlaylistDefinitions	PlaylistDefinitions	1.0
PlaylistElement	PlaylistElement	1.0
PlaylistMembership	Deprecated.	1.0
PlaylistMemberships	Deprecated.	1.0
RepositoryPath	RepositoryPath	1.0
RepositoryPaths	RepositoryPaths	1.10
ShareAssociation	ShareAssociation	1.10
SharedType	SharedType	1.0
TaggedContact	TaggedContact	1.14
TaggedContacts	TaggedContacts	1.14
Track	Playlist track.	1.13
TrackPlaylistDefinition	Playlist definition containing media provider tracks.	1.13
TrackPlaylistDefinitions	List of playlists containing media provider tracks.	1.13
Tracks	Playlist tracks.	1.13
UserDetails	UserDetails	1.0
ViewMember	ViewMember	1.0

A.4. Attribute

Since:1.0

Table A.3. Attribute

Field	Type	Description	Cardinality	Since
@name	xs:string	xs:string	1..1	1.0

Referenced by: **PlaylistDefinition PlaylistElement**

A.5. AttributedRepositoryPath

Since:1.0

Table A.4. AttributedRepositoryPath

Field	Type	Description	Cardinality	Since
@name	xs:string	xs:string	0..1	1.0

Referenced by: **PlaylistDefinition**

A.6. Checksum

Since:1.0

Referenced by: **PlaylistElement**

A.7. InvalidPlaylistElement

Since:1.0

Table A.5. InvalidPlaylistElement

Field	Type	Description	Cardinality	Since
link	**Link**		0..*	1.0
pathId	xs:string	xs:string	1..1	1.0

Referenced by: **Playlist**

A.8. Link

Since:1.0

Table A.6. Link

Field	Type	Description	Cardinality	Since
@rel	xs:string	xs:string	0..1	1.0

Referenced by: **InvalidPlaylistElement** **Linkable** **PagedList** **Playlist** **PlaylistDefinition** **PlaylistDefinitions** **PlaylistElement** **PlaylistMembership** **PlaylistMemberships** **Track** **TrackPlaylistDefinition** **TrackPlaylistDefinitions** **Tracks** **UserDetails**

A.9. Linkable

Since:1.0

Table A.7. Linkable

Field	Type	Description	Cardinality	Since
link	**Link**		0..*	1.0

A.10. PagedList

Since:1.0

Table A.8. PagedList

Field	Type	Description	Cardinality	Since
link	**Link**		0..*	1.0
@start	xs:int	xs:int	0..1	1.0
@total	xs:int	xs:int	1..1	1.0

A.11. Playlist

Since:1.0

Table A.9. Playlist

Field	Type	Description	Cardinality	Since
link	**Link**		0..*	1.0
uri	xs:anyURI	Unique identifier of the resource.	0..1	1.9
playlistElement	**PlaylistElement**		0..*	1.0
invalidPlaylistElement	**InvalidPlaylistElement**		0..*	1.0
@totalCount	xs:int	xs:int	0..1	1.0
@pathCount	xs:int	xs:int	0..1	1.0

A.12. PlaylistDefinition

Since:1.0

Table A.10. PlaylistDefinition

Field	Type	Description	Cardinality	Since
link	**Link**		0..*	1.0

Field	Type	Description	Cardinality	Since
uid	xs:string	A unique identifier for the playlist.	0..1	1.0
owner	**UserDetails**	The owner of the playlist.	0..1	1.0
name	xs:string	The name of the playlist.	1..1	1.0
type	xs:string	The type of the playlist. Can be one of 'image', 'music', 'video' or 'image-video'.	1..1	1.0
mimeTypeRegex	xs:string		0..1	1.0
attributedRepositoryPath	**AttributedRepositoryPath**		0..*	1.0
repositoryPath	**RepositoryPath**	The path to a file in the playlist. Must include the repository name, for example, desktop:/myphoto.jpg.	0..*	1.0
email	xs:string	Deprecated. Vault Extensions no longer handles the sharing of playlists with other users. Use the Cloud Share component instead.	0..*	1.0
msisdn	xs:string	Deprecated. Vault Extensions no longer handles the sharing of playlists with other users.	0..*	1.0
expirationDate	xs:dateTime		0..1	1.0
expired	xs:boolean		0..1	1.0
creationDate	xs:dateTime		0..1	1.0
lastModifiedDate	xs:dateTime	The date and time the playlist was last modified. Updated when content is added to, or removed from, the playlist.	0..1	1.10
invitationDate	xs:dateTime	Deprecated. Vault Extensions no longer handles the sharing of playlists with other users. Use the Cloud Share component instead.	0..1	1.0
public	xs:boolean	Deprecated. Playlists can no longer be shared publicly.	0..1	1.0
publicUrl	xs:string	Deprecated. Playlists can no longer be shared publicly.	0..1	1.0
clientAttribute	**Attribute**	Client specified attribute for the playlist.	0..*	1.0
uri	xs:anyURI	Unique identifier of the resource.	0..1	1.0
systemAttribute	**Attribute**	System attribute for the playlist. Can take a 'name' attribute of 'Category', with a value of either 'SmartAlbum' or 'SmartFolder'.	0..*	1.10
validPathCount	xs:int	Number of valid entries in the playlist. Valid entries exclude playlist entries that have been deleted.	0..1	1.10
shareAssociation	**ShareAssociation**	If present, indicates that the playlist is shared. Contains the share details.	0..*	1.10

Referenced by: **PlaylistDefinitions**

A.13. PlaylistDefinitions

Since:1.0

Table A.11. PlaylistDefinitions

Field	Type	Description	Cardinality	Since
link	**Link**		0..*	1.0
playlistDefinition	**PlaylistDefinition**		0..*	1.0
@totalCount	xs:int	xs:int	0..1	1.0

A.14. PlaylistElement

Since:1.0

Table A.12. PlaylistElement

Field	Type	Description	Cardinality	Since
link	**Link**		0..*	1.0
filename	xs:string	xs:string	1..1	1.0
repositoryPath	xs:string	xs:string	0..1	1.0
mimeType	xs:string	xs:string	1..1	1.0
contentPermission	xs:string	xs:string	1..1	1.0
size	xs:long	xs:long	1..1	1.0
versionCreated	xs:dateTime	xs:dateTime	1..1	1.0
creationDate	xs:string	When the file was created.	0..1	1.0
pathId	xs:string	An identifier used in ordering and reordering elements in the playlist.	1..1	1.0
fileAttribute	**Attribute**		0..*	1.0
contentToken	xs:string	A unique token that identifies a file.	0..1	1.0
checksum	**Checksum**		0..1	1.0
contentAccessible	xs:boolean		0..1	1.0
extension	xs:string	The file extension	0..1	1.0
tags	xs:string		0..1	1.0
title	xs:string	Track title	0..1	1.0
artist	xs:string	Artist name	0..1	1.0
year	xs:string	Recording year	0..1	1.0
genre	xs:string	Musical genre	0..1	1.0
album	xs:string	Album name	0..1	1.0
track	xs:string	Track number	0..1	1.0
viewUid	**ViewMember**		0..*	1.0
uri	xs:anyURI	Unique identifier of the resource.	0..1	1.9
systemAttribute	**Attribute**	Possible system attributes include Capture-Date, Geo-Latitude, Geo-Longitude, Height, Weight, Width, Location-Street, Location-Locality, Location-County, Location-Region, Location-PostalCode, Location-Country, Location-CountryCode and Tags. However, a subset of these attributes can be defined in server configuration.	0..*	1.0

Field	Type	Description	Cardinality	Since
taggedContacts	**TaggedContacts**	Tagged contacts for this item.	0..1	1.14

Referenced by: **Playlist**

A.15. PlaylistMembership

Deprecated. Playlists no longer support the concept of 'membership'.

Since:1.0

Table A.13. PlaylistMembership

Field	Type	Description	Cardinality	Since
link	**Link**		0..*	1.0
uid	xs:string	xs:string	0..1	1.0
owner	**UserDetails**		1..1	1.0
name	xs:string	xs:string	1..1	1.0
type	xs:string	xs:string	1..1	1.0
mimeTypeRegex	xs:string	xs:string	0..1	1.0
expirationDate	xs:dateTime	xs:dateTime	0..1	1.0
expired	xs:boolean	xs:boolean	0..1	1.0
inviteKey	xs:string	xs:string	0..1	1.0
password	xs:string	xs:string	0..1	1.0
creationDate	xs:dateTime	xs:dateTime	0..1	1.0
invitationDate	xs:dateTime	xs:dateTime	0..1	1.0
subscribed	xs:boolean	xs:boolean	1..1	1.0
@pathCount	xs:int	xs:int	0..1	1.0

Referenced by: **PlaylistMemberships**

A.16. PlaylistMemberships

Deprecated. Playlists no longer support the concept of 'membership'.

Since:1.0

Table A.14. PlaylistMemberships

Field	Type	Description	Cardinality	Since
link	**Link**		0..*	1.0
playlistMembership	**PlaylistMembership**		0..*	1.0
@totalCount	xs:int	xs:int	0..1	1.0

A.17. RepositoryPath

Since:1.0

Table A.15. RepositoryPath

Field	Type	Description	Cardinality	Since
@pathId	xs:string	xs:string	0..1	1.0

Referenced by: **PlaylistDefinition RepositoryPaths**

A.18. RepositoryPaths

Since:1.10

Table A.16. RepositoryPaths

Field	Type	Description	Cardinality	Since
repositoryPath	**RepositoryPath**		0..*	1.0

A.19. ShareAssociation

Since:1.10

Table A.17. ShareAssociation

Field	Type	Description	Cardinality	Since
@uid	xs:string	The unique ID of the share.	0..1	1.0
@visibility	xs:string	The share visibility. Can be one of 'private' or 'public'.	0..1	1.0
@name	xs:string	The name of the share.	0..1	1.0

Referenced by: **PlaylistDefinition**

A.20. SharedType

Since:1.0

Table A.18. SharedType

Value	Description
root	root
descendant	descendant

Referenced by: **ViewMember**

A.21. TaggedContact

Since:1.14

Table A.19. TaggedContact

Field	Type	Description	Cardinality	Since
id	xs:string	Unique identifier of the tagged contact.	0..1	1.0
name	xs:string	Name of the tagged contact.	0..1	1.0
coordinates	xs:string	X Y co-ordinates for tagged contact.	1..1	1.0
info	xs:string	Contains extra information about tagged contact.	0..1	1.0

Referenced by: **TaggedContacts**

A.22. TaggedContacts

Since:1.14

Table A.20. TaggedContacts

Field	Type	Description	Cardinality	Since
taggedContact	**TaggedContact**		0..*	1.0

Referenced by: **PlaylistElement**

A.23. Track

Playlist track.

Since:1.13

Table A.21. Track

Field	Type	Description	Cardinality	Since
link	**Link**		0..*	1.0
id	xs:string	xs:string	1..1	1.0
title	xs:string	xs:string	1..1	1.0
artist	xs:string	xs:string	0..1	1.0
album	xs:string	xs:string	0..1	1.0
albumArtist	xs:string	xs:string	0..1	1.0
genre	xs:string	xs:string	0..1	1.0
year	xs:int	xs:int	0..1	1.0
compilation	xs:boolean	xs:boolean	0..1	1.0
track	xs:int	xs:int	0..1	1.0

Referenced by: **Tracks**

A.24. TrackPlaylistDefinition

Playlist definition containing media provider tracks.

Since:1.13

Table A.22. TrackPlaylistDefinition

Field	Type	Description	Cardinality	Since
link	**Link**		0..*	1.0
uid	xs:string	xs:string	1..1	1.0
name	xs:string	xs:string	1..1	1.0
trackCount	xs:int	xs:int	0..1	1.0

Referenced by: **TrackPlaylistDefinitions**

A.25. TrackPlaylistDefinitions

List of playlists containing media provider tracks.

Since:1.13

Table A.23. TrackPlaylistDefinitions

Field	Type	Description	Cardinality	Since
link	**Link**		0..*	1.0
trackPlaylistDefinition	**TrackPlaylistDefinition**		0..*	1.0
@totalCount	xs:int	xs:int	0..1	1.0

A.26. Tracks

Playlist tracks.

Since:1.13

Table A.24. Tracks

Field	Type	Description	Cardinality	Since
link	**Link**		0..*	1.0
track	**Track**		0..*	1.0
@totalCount	xs:long	xs:long	0..1	1.0

A.27. UserDetails

Since:1.0

Table A.25. UserDetails

Field	Type	Description	Cardinality	Since
link	**Link**		0..*	1.0
phone	xs:string	xs:string	0..1	1.0
email	xs:string	xs:string	0..1	1.0

Field	Type	Description	Cardinality	Since
firstName	xs:string	xs:string	0..1	1.0
lastName	xs:string	xs:string	0..1	1.0

Referenced by: **PlaylistDefinition PlaylistMembership**

A.28. ViewMember

Since:1.0

Table A.26. ViewMember

Field	Type	Description	Cardinality	Since
@type	**SharedType**		0..1	1.0
@viewType	xs:string	xs:string	0..1	1.0
@publicUrl	xs:string	xs:string	0..1	1.0

Referenced by: **PlaylistElement**

B

Digital Vault Extensions Errors

B.1. Error Codes

As well as standard HTTP error response codes, the Digital Vault Extensions API server can return the following error codes and messages in the body of responses:

Table B.1. Digital Vault Extensions Error Codes

Digital Vault Extensions Error Code	HTTP Error Code	Digital Vault Extensions Error Message	Cause
2219		The number of repository paths to be listed must not exceed the limit of *`{path limit}`*	The client exceeded the maximum allowed number of `repositoryPath` parameters when calling the **dvx.user.folder.list (Deprecated)** API. The limit is 10.
3000	400	Invalid playlist format	The client tried to create a playlist in an invalid format, for example, by including a `clientAttribute` that exceeded the maximum character limit.
3001		The playlist element *`{filename}`* with pathId *`{path ID`* has no matching entry in playlist definition. This is not legal.	Non-matching playlist and playlist definition
3002	400	Invalid playlist name	The client tried to create a playlist with an invalid character in the name.
3003		Invalid system attribute name *`{attributeName}`*	The client supplied an invalid system attribute name in the playlistDefinition object in the body of the request. Only one system attribute is currentlly supported, namely `category` which can take a value of `SmartAlbum` or `SmartFolder`.
3004		Invalid path ID *`{pathId}`*	
3100	413	Too many elements in bulk request	The client has included too many files in a call to the **dvx.user.repository.file.bulk.create** API
4000			The client attempted to access playlist membership APIs with API version 1.9 and above.
4001	400		The client tried to share a playlist even though the sharing feature is not supported with API version 1.9 and above. For example, MSISDN must be removed from the model being sent.