



Tape, REST API and more

XRootD and FTS Workshop 2023 at JSI

João Pedro Lopes on behalf of the FTS team

Monday 27th March 2023

Introduction

Tape REST API, why?

- Provide a modern, uniform way of managing tape data movements across all tape storage implementations in WLCG
- Unify tape and transfer management under the same protocol
 - Replace SRM for tape data management with a simpler interface

How?

- Storage (EOSCTA, dCache, StoRM) + FTS/Gfal2 developers involved
- Tape REST API Reference document [v1, May 2022]
- DOMA taskforce:
wlcg-tape-rest-api-discussions@cern.ch.

WLCG Tape REST API (v1) reference document

Introduction

This document contains the WLCG tape REST API specifications created in collaboration by EOS+CTA (<https://cta.web.cern.ch/cta/>), dCache (<https://www.dcache.org/>) and StoRM (<https://italiangrid.github.io/storm/>).

The FTS project (<https://fts.web.cern.ch/fts/>) was also involved as the main client for the REST API.

In this document, the requirements level will be indicated following the RFC 2119 (<https://datatracker.ietf.org/doc/html/rfc2119>).

Main purpose of the WLCG tape REST API

The WLCG tape REST API offers a common interface allowing clients to manage disk residency of tape stored files and observe the progress of files as they are written to tape.

The protocol was designed to be implementation agnostic for the common tape-backed storage systems in WLCG: the same protocol is used to access a dCache, StoRM or EOS+CTA tape storage system.

Bulk operations provide an optimization to allow client and server to work efficiently with large numbers of files.

Set of operations provided by the API

Functionality	Description
STAGE	Request that tape-stored files are made available with disk latency.
RELEASE	Indicate that files previously staged through STAGE are no longer required with disk latency.
ABORT	Request information about the operation.

[Tape REST API reference document \[May 2022\]](#)

Outline

- I. FTS/Gfal2 status**
- II. Tape REST API Resources**
- III. Archive a file to tape**
- IV. Recall the file from tape**
- V. Release file from disk**

I. Tape REST API, FTS/Gfal2 status

- **All** FTS instances at CERN support the Tape REST API
- Gfal2 v2.21.0: released, full support
- FTS v3.12.2: release, full support
- Newer patch releases include better logging and error reporting (good to stay on the latest version)
- RPM's at:
 - <https://fts-repo.web.cern.ch/fts-repo/fts3-prod-el7.repo>
 - <https://dmc-repo.web.cern.ch/dmc-repo/dmc-el7.repo>

I. Tape REST API, FTS/Gfal2 status

- Successful file archival using the Tape REST API at eosctalhcb.cern.ch by the LHCb experiment

VO: lhcb

Delegation ID: 32d8393f6eee880a
Submitted time: 2023-03-26T21:37:15Z
Job finished: 2023-03-26T21:55:19Z
Priority: 3
Bring online: -1
Archive timeout: 86400

Received by fts-lhcb-003.cern.ch
Job expires:
Overwrite flag: Y
Job type: N
Cancel flag:
Pin lifetime: -1
Target QoS:

Metadata:

```
{"operationID": 5821312, "rmsReqID": 40223861, "sourceSE": "IN2P3_MC-DST", "targetSE": "CERN-ARCHIVE", "auth_method": "certificate"}
```

Files transferred	Bytes transferred	Submission time	Start time	Running time	Avg. file throughput	Current job throughput
1 out of 1	1.58 GiB	2023-03-26T21:37:15Z	2023-03-26T21:37:19Z (+4s)	5 s	333.68 MiB/s	-

Showing 1 to 1 out of 1

SUBMITTED DELETE READY STAGING ARCHIVING ACTIVE STARTED CANCELED FAILED 1 FINISHED NOT_USED

First Previous 1 Next Last

File ID	File State	File Size	Throughput	Remaining	Start Time	Finish Time	Staging Start	Staging End	Archiving Start	Archiving End
+ 35224842	FINISHED	1.58 GiB	333.68 MiB/s	-	2023-03-26T21:37:15Z	2023-03-26T21:37:19Z	-	-	2023-03-26T21:38:00Z	2023-03-26T21:55:19Z

https://ccdavlhcb.in2p3.fr:2880/pnfs/in2p3.fr/data/lhcb/LHCb-Disk/lhcb/MC/2016/DST/00179555/0000/00179555_00000353_6.dst

https://eosctalhcb.cern.ch:8444/eos/ctalhcb/archive/grid/lhcb/archive/lhcb/MC/2016/DST/00179555/0000/00179555_00000353_6.dst

Outline

- I. FTS/Gfal2 status
- II. Tape REST API Resources**
- III. Archive a file to tape
- IV. Recall the file from tape
- V. Release file from disk

II. Tape REST API Resources

Discovery endpoint:

- Introduces a discovery mechanism: ``.well-known/wlwg-tape-rest-api``
 - “... allows the client to discover metadata about the service”
 - “The well-known endpoint is found by taking the WebDAV endpoint and resolving the absolute path `/.well-known/wlwg-tape-rest-api`”
 - “Conforming servers MUST support the discovery endpoint ...”
- First request that Gfal2/FTS when interacting with a Tape REST API endpoint:

```
[root@joaopedro ~]# davix-get --cert /tmp/x509up_u0 --capath /etc/grid-security/certificates/  
https://eosctaatlaspps.cern.ch:8444/.well-known/wlwg-tape-rest-api | jq .  
{  
  "endpoints": [  
    {  
      "uri": "https://eosctaatlaspps.cern.ch:8444/api/v1",  
      "version": "v1"  
    }  
  ],  
  "sitename": "cern-tape-archive-eosctaatlaspps"  
}
```

II. Tape REST API Resources

Operations provided by the API:

- Only a very minimalistic set of operations are provided
- From the [Tape REST API reference document](#):

Functionality	Description
STAGE	Request that tape-stored files are made available with disk latency.
RELEASE	Indicate that files previously staged through STAGE are no longer required with disk latency.
ARCHIVEINFO	Request information about the progression of writing files to tape

Outline

- I. FTS/Gfal2 status
- II. Tape REST API Resources
- III. Archive a file to tape**
- IV. Recall the file from tape
- V. Release file from disk

III. Archive a file to tape

Tracking the file archiving progress:

- Use the Tape REST API **/archiveinfo** resource to monitor disk/tape residency
 - “Request information about the progression of writing files to tape”

```
[root@joaopedro ~]# echo $JSON | jq
{
  "paths": [
    "/eos/ctaataspps/ftstest/IBM1L9/ftstest1/1/test01001000"
  ]
}
[root@joaopedro ~]# davix-http -X POST --cert /tmp/x509up_u0 --capath /etc/grid-security/certificates/ -H
"Content-Type: application/json" https://eosctaataspps.cern.ch:8444/api/v1/archiveinfo/ --data $JSON | jq
[
  {
    "locality": "TAPE",
    "path": "/eos/ctaataspps/ftstest/IBM1L9/ftstest1/1/test01001000"
  }
]
```

III. Archive a file to tape

Tracking the file archiving progress:

- Also possible to monitor archival of a file using Gfal2 CLI directly:

```
[root@joaopedro ~]# gfal-archivepoll https://eosctaataspps.cern.ch:8444/eos/ctaataspps/ftstest/IBM1L9/ftstest1/1/test01001000 -vv
INFO [gfal_module_load] plugin /usr/lib64/gfal2-plugins//libgfal_plugin_http.so loaded with success
INFO Davix: > GET /.well-known/wlwg-tape-rest-api HTTP/1.1
> User-Agent: gfal2-util/1.8.0 gfal2/2.21.4 neon/0.0.29
> Keep-Alive:
> Connection: Keep-Alive
> TE: trailers
> Host: eosctaataspps.cern.ch:8444
>
INFO Davix: < HTTP/1.1 200 OK
INFO Davix: < Connection: Keep-Alive
INFO Davix: < Content-Length: 160
INFO Davix: < Date: Sun, 26 Mar 2023 19:05:04 GMT
INFO Davix: < application/type: json
INFO Davix: <
INFO Davix: > POST /api/v1/archiveinfo/ HTTP/1.1
> User-Agent: gfal2-util/1.8.0 gfal2/2.21.4 neon/0.0.29
> TE: trailers
> Host: eosctaataspps.cern.ch:8444
> Content-Type: application/json
> Content-Length: 70
>
INFO Davix: < HTTP/1.1 200 OK
INFO Davix: < Connection: Keep-Alive
INFO Davix: < Content-Length: 101
INFO Davix: < Date: Sun, 26 Mar 2023 19:05:04 GMT
INFO Davix: < application/type: json
INFO Davix: <
https://eosctaataspps.cern.ch:8444/eos/ctaataspps/ftstest/IBM1L9/ftstest1/1/test01001000 READY
```

III. Archive a file to tape

Archiving of a file:

- The [Tape REST API reference document](#) specifies the following six different values for the file locality:
 - FTS will only mark “archiving transfers” as *FINISHED* when the locality is “**TAPE**” or “**DISK_AND_TAPE**”

Locality	Semantic
DISK	Indicates that the file's data is located only on disk storage.
TAPE	Indicates that the file's data is located only on a tape storage system.
DISK_AND_TAPE	Indicates that the file's data is located on a tape as well as on a disk storage system.
LOST	Indicates the file's data is considered unrecoverable.
NONE	May be used if the file is empty (0 size)
UNAVAILABLE	Indicates that the file's data is unavailable due to a temporary hardware failure, disk server being rebooted, etc.

Outline

- I. FTS/Gfal2 status
- II. Tape REST API Resources
- III. Archive a file to tape
- IV. Recall the file from tape**
- V. Release file from disk

IV. Recall the file from tape - Submission

Submit a “stage” request:

- Use the Tape REST API **/stage** resource to request the endpoint to write the file to disk
 - “Request that tape-stored files are made available with disk latency”

```
[root@joaopedro ~]# echo $JSON | jq
{
  "files": [
    {
      "path": "/eos/ctaataspps/ftstest/IBM1L9/ftstest1/1/test01001000"
    }
  ]
}
[root@joaopedro ~]# davix-http -X POST --cert /tmp/x509up_u0 --capath /etc/grid-security/certificates/ -H
"Content-Type: application/json" https://eosctaataspps.cern.ch:8444/api/v1/stage/ --data $JSON | jq
{
  "requestId": "aa21c332-cc0a-11ed-a5ab-ac1f6b4cd618"
}
```

IV. Recall the file from tape - Track the progress

Track the progress of a “stage” request:

- Use the Tape REST API `/stage/<request_id>` resource to track the progress of the request
 - File is not yet on disk

```
[root@joaopedro ~]# davix-http -X GET --cert /tmp/x509up_u0 --capath /etc/grid-security/certificates/ -H
"Content-Type: application/json" https://eosctaataspps.cern.ch:8444/api/v1/stage/aa21c332-cc0a-11ed-a5ab-
-ac1f6b4cd618 | jq
{
  "createdAt": 1679858169,
  "files": [
    {
      "onDisk": false,
      "path": "/eos/ctaataspps/ftstest/IBM1L9/ftstest1/1/test01001000"
    }
  ],
  "id": "aa21c332-cc0a-11ed-a5ab-ac1f6b4cd618",
  "startedAt": 1679858169
}
```


IV. Recall the file from tape - Track the progress

Track the progress of a “stage” request:

- File has been written to disk

```
[root@joaopedro ~]# davix-http -X GET --cert /tmp/x509up_u0 --capath /etc/grid-security/certificates/ -H
"Content-Type: application/json" https://eosctaataspps.cern.ch:8444/api/v1/stage/aa21c332-cc0a-11ed-a5
ab-ac1f6b4cd618 | jq
{
  "createdAt": 1679858169,
  "files": [
    {
      "onDisk": true,
      "path": "/eos/ctaataspps/ftstest/IBM1L9/ftstest1/1/test01001000"
    }
  ],
  "id": "aa21c332-cc0a-11ed-a5ab-ac1f6b4cd618",
  "startedAt": 1679858169
}
```

- FTS will only start the copy when the “*onDisk*” field is *true*

IV. Recall the file from tape - Cancel the request

Cancel the “stage” request:

- Use the Tape REST API `/stage/<request_id>/cancel` resource to cancel the stage request
- From the [tape REST API reference document](#):
 - “For each cancelled file, the endpoint MAY cancel any staging activity”

```
[root@joaopedro ~]# echo $JSON | jq .
{
  "paths": [
    "/eos/ctaataspps/ftstest/IBM1L9/ftstest1/1/test01000001"
  ]
}
[root@joaopedro ~]# davix-http -X POST --cert /tmp/x509up_u0 --capath /etc/grid-security/certificates/ -H
"Content-Type: application/json" https://eosctaataspps.cern.ch:8444/api/v1/stage/5d280c36-cc0e-11ed-a11
8-ac1f6b4cd618/cancel --data $JSON | jq
```

- Client does not know if the *cancel* request was honoured on the server side

Outline

- I. FTS/Gfal2 status
- II. Tape REST API Resources
- III. Archive a file to tape
- IV. Recall the file from tape
- V. Release file from disk**

V. Release file from disk

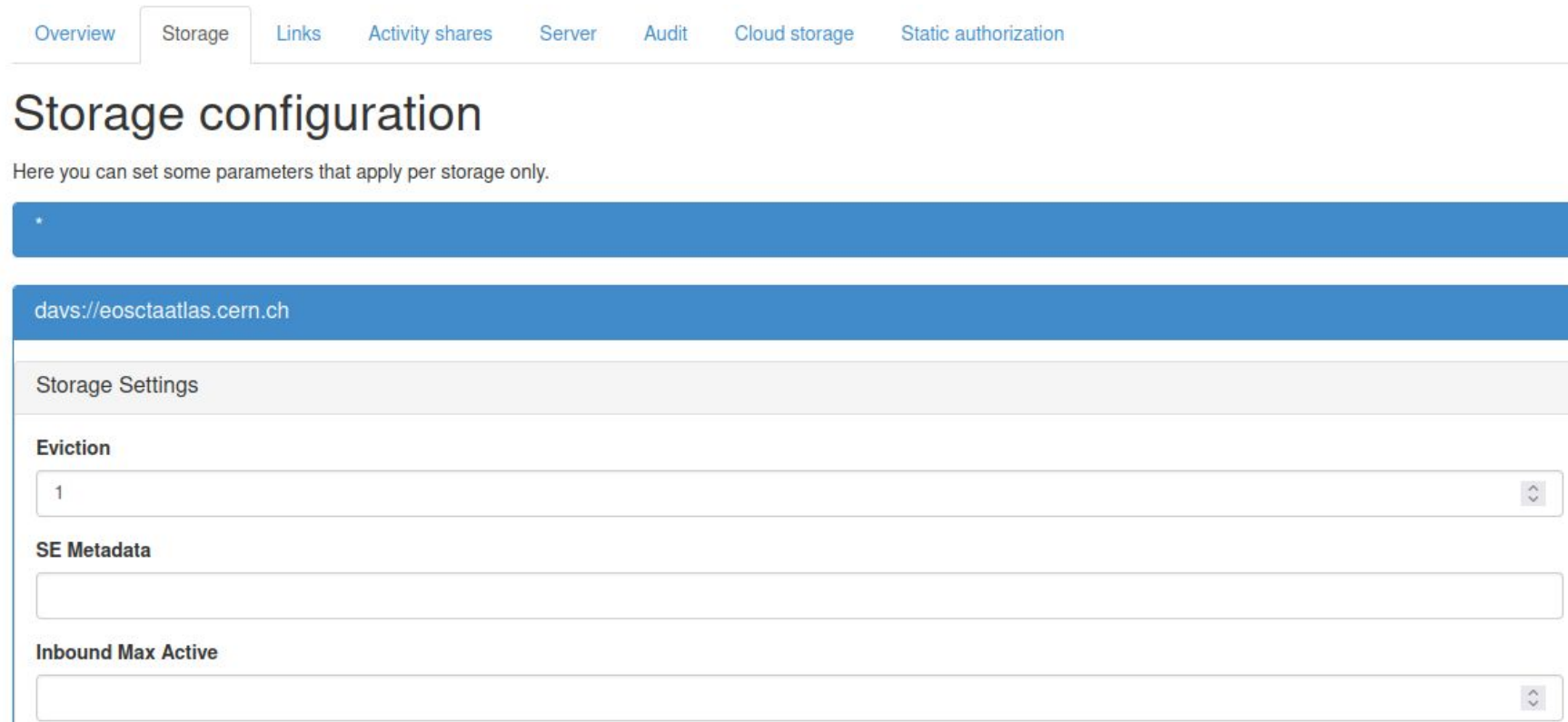
Delete the recalled file from the disk area after it has been copied out:

- Use the Tape REST API **/release** resource to “evict” files from disk
- Files are never deleted from tape after a request to **/release**
- From the [Tape REST API reference document](#):
 - “The storage provider MAY remove these files from the disk cache”

```
[root@joaopedro ~]# echo $JSON | jq .
{
  "paths": [
    "/eos/ctaataspps/ftstest/IBM1L9/ftstest1/1/test01000001"
  ]
}
[root@joaopedro ~]# davix-http -X POST --cert /tmp/x509up_u0 --capath /etc/grid-security/certificates/ -H
"Content-Type: application/json" https://eosctaataspps.cern.ch:8444/api/v1/release/5d280c36-cc0e-11ed-a
118-ac1f6b4cd618 --data $JSON | jq
[root@joaopedro ~]#
```

V. Release file from disk: Configuring FTS

- EOSCTA (+ Antares) storage endpoints use a small and fast “disk-buffer” as a retrieve area
- Once files are copied to the final destination the disk replica should be “evicted” (i.e released)
- FTS can be configured to “evict” files from storage endpoints after successful copies
 - Example from <https://fts3-atlas.cern.ch:8446/config/se>:



The screenshot shows a web interface for configuring storage settings. At the top, there are navigation tabs: Overview, Storage (selected), Links, Activity shares, Server, Audit, Cloud storage, and Static authorization. Below the tabs is the title "Storage configuration" and a subtitle "Here you can set some parameters that apply per storage only." The main content area is titled "Storage Settings" and contains three sections: "Eviction" with a dropdown menu set to "1", "SE Metadata" with an empty text input field, and "Inbound Max Active" with a dropdown menu.

Conclusions

- FTS/Gfal2 fully support Tape REST API (v1)
- Some experiments start to using it in production
- Waiting for more storage elements to provide a Tape REST API endpoint
 - Gfal2 and Davix can be used to validate/debug before submitting jobs to FTS



home.cern