Statement Deletion HTTP Interfaces

Authorization

When using this interface, you must additionally supply your Basic Auth details with each request in the `Authorization` header. Your Basic Auth details can be found under **Settings** > **Clients**.

The deletion APIs require that the "Delete statements" (`'statements/delete'`) scope is enabled on the client. If a client also has a store (`lrs_id`) attached, this will be used to further filter down deletions only to this store.

Single Statement Deletion

This leverages the existing REST API for statements.

```
DELETE https://jisc.learninglocker.net/api/v2/statement/111aaa1111a111111aa11112
Authorization: Basic YOUR_BASIC_AUTH
```

A request like the one above, will respond with a 204 response like the one below.

```
HTTP/1.1 204 NO CONTENT
```

The statement is deleted immediately and the 204 represents a receipt of the deletion event succeeding in the database.

Batch Statement Deletion

These endpoints allow you to send in deletion jobs to be processed, as well as stop a batch deletion job.

As the filter passed in may apply to a large amount of data, the batch delete job is split out into batches, each deleting up to 1000 records at a time. Each successive batch will trigger another deletion job to the worker, until no more statements exist in the database matching that filter.

Initialising a batch deletion

Sending a POST with a JSON body holding the required deletion filter to the following endpoint will create a job to remove all data matching that filter from the respective organisation (or store) that the client is attached to.

e.g. Deletes all completions in the client's organisation or store

```
POST https://jisc.learninglocker.net/api/v2/batchdelete/initialise
Authorization: Basic YOUR_BASIC_AUTH
Content-Type: application/json; charset=utf-8
```

```
{
  "filter": {
    "statement.verb.id": "http://adlnet.gov/expapi/verbs/completed"
  }
}
```

If the client used to make the request also has a store (`lrs_id`) attached, this will be used to further filter down deletions only to this store.

An intialise request will return a 200 HTTP response with a JSON version of the batch deletion job (see Schema)

Terminating batch deletions

You may choose to terminate one or all batch deletions for an organisation using the following commands.

Note that if a deletion is currently in progress and a worker job to delete a batch has been issued, up to 1000 records may be deleted before the termination command is respected.

This is due to how we batch the deletions into blocks of 1000. Once a batch has started, it cannot be stopped without manually stopping the Worker's Node process running the deletion job. However, no subsequent batches will be processed once that batch has finished.

Terminating a single batch deletion

Stop a specific batch deletion from executing any more batches.

```
GET
https://jisc.learninglocker.net/api/v2/batchdelete/terminate/111aaa1111a111111aa11
112
Authorization: Basic YOUR_BASIC_AUTH
```

Terminating all batch deletions

Stop all batch deletions from executing any more batches

```
GET https://jisc.learninglocker.net/api/v2/batchdelete/terminate/all
Authorization: Basic YOUR_BASIC_AUTH
```

Viewing batch deletions

Schema

Name	Description
_id	The id of the batch delete job.

Name	Description
organisation	The id of the organisation that this job belongs to.
filter	A stringified JSON Mongo query - records matching this filter are deleted
pageSize	Total records deleted per batch (defaults to 1000, no way to customise outside of code change and rebuild)
deleteCount	How many records have been deleted so far
total	Total number of statements found for deletion at initialise
processing	Boolean value; is there a deletion batch currently being actioned
done	Boolean value; has the job finished or been terminated
createdAt	When this document was created
updatedAt	When this document was last updated

_The `filter` field is stored as text to account for `.` (dot) characters used in query keys - these are invalid in Mongo JSON structures.

Example

```
{
  "_id": "111aaa1111a111111aa11111",
  "organisation": "111aaa1111a111111aa11111",
  "filter": "{\"statement.verb.id\":
\"http://adlnet.gov/expapi/verbs/completed\"}",
  "pageSize": 1000,
  "deleteCount": 100000,
  "total": 100000,
  "processing": false,
  "done": true,
  "createdAt": "2019-01-01T00:00:00Z",
  "updatedAt": "2019-01-01T00:00:00Z"
}
```

The Batch Delete model may be retrieved using the GET REST or Connection APIs but other HTTP methods are disabled (PUT, PATCH, DELETE) and are instead replaced by the initialise and terminate routes specified above.

Examples: (using the Connection API)

Note; query parameters should be URL encoded - these examples have not been, for readability

Fetch a particular job by `_id`

```
GET https://jisc.learninglocker.net/api/connection/batchdelete?filter={"_id":
{"$oid":"111aaa1111a111111aa11112"}}
```

```
Authorization: Basic YOUR_BASIC_AUTH
```

Fetch the 5 most recently completed/terminated jobs

```
GET https://jisc.learninglocker.net/api/connection/batchdelete?filter=
{"done":true}&sort={"updatedAt":-1, "_id": 1}&first=5
Authorization: Basic YOUR_BASIC_AUTH
```

Fetch the 5 most recently created and unfinished jobs

```
GET https://jisc.learninglocker.net/api/connection/batchdelete?filter=
{"done":false}&sort={"createdAt":-1, "_id": 1}&first=5
Authorization: Basic YOUR_BASIC_AUTH
```