API for the writeLaTeX-Git Bridge

JLM

April 10, 2024

1 Fetching a Project from WriteLaTeX

There are three API calls that will likely be of interest. You can run them against this server, radiant-wind-3058.herokuapp.com, but they're not on the production server yet.

1.1 Get Doc

A “doc” is our internal term for a “project”. This endpoint returns the latest version id, when the latest version was created (ISO8601), and the user that last edited the project (if any, otherwise null).

```
GET https://.../api/v0/docs/1826rqgsdb
# => {
   ”latestVerId”: 39,
   ”latestVerAt”: ”2014-11-30T18:35:27Z”,
   ”latestVerBy”: {
    ”email”: ”jdleesmiller@gmail.com”,
    ”name”: ”John Lees-Miller”
   }}
```

1.2 Get Saved Vers

A “saved ver” is a version of a doc, saved by via the versions menu. Note that this query is not currently paginated.

```
GET https://.../api/v0/docs/1826rqgsdb/saved_vers
# => [
   {”versionId”:39,
   ”comment”:”with more files”,
   ”user”:{
    ”email”:”jdleesmiller@gmail.com”,
    ”name”:”John Lees-Miller”},
   ”createdAt”:”2014-11-05T18:02:19Z”},
   {”versionId”:24,
   ”comment”:”first draft”,
   ”user”:{
    ”email”:”jdleesmiller@gmail.com”,
    ”name”:”John Lees-Miller”},
   ”createdAt”:”2014-11-05T17:56:58Z”}]
```

1.3 Get Snapshot for Version

A snapshot contains the content of a project in the given version. You can safely request a snapshot of any version that is, or was at any point in the last 24 hours, (1) a saved version, or (2) the current version. (Older versions may or may not have been moved to cold storage.)

The srcs array contains (content, file name) pairs; the atts array contains (URL, file name) pairs.

```
GET https://.../api/v0/docs/1826rqgsdb/snapshots/39
# => {
  "srcs":[
   ["This text is from another file.","foo/bar/servermock.tex"],
   ["\\documentclass[a4paper]{article}\n...","main.tex"]],
  "atts":[
   ["https://writelatex-staging.s3.amazonaws.com/filepicker/1
      ENnu6zJSGyslI3DuNZD_min_mean_wait_evm_7.eps.150dpi.png","
      min_mean_wait_evm_7_eps_150dpi.png"]]}
```

2 Pushing a Project to WriteLaTeX

2.1 Push a Project

```
# NB: JLM originally said PUT, but he now thinks POST is better
# NB: you must set a Content-Type: application/json header for this request
# in order to specify the data as JSON in the request body
POST https://.../api/v0/docs/1826rqgsdb/snapshots
Data:
{
  latestVerId: integer,
  files: [
   {
    name: string path (forward slashes, relative to root)
    url: string (but only if the file is modified; else no url given)
   }, ...
  ]
  postbackUrl: url to post result back to
}
Response on success:
{
  status: 202,
  code: "accepted",
  message: "Accepted"
}
Response on out of date:
{
  status: 409, # Conflict
  code: "outOfDate",
  message: "Out of Date"
}

Postback Data (METHOD POST):
On success:
{
  code: "upToDate",
  latestVerId: integer
}
On out of date:
{
  code: "outOfDate",
  message: "Out of Date"
}
On error with the files list (e.g. file extension not allowed):
{
  code: "invalidFiles",
  errors: [ {
   file: the file name from the snapshot,
   state: "error"|"disallowed"|"unclean_name"
  }, ... ]
}
If the file's error state is unclean_name, the error object will
    alsocontain a property cleanFile that contains the name of the file
    after it has been "cleaned" to meet our file naming requirements; for
    other file error states, this property is not present.
On error with the project as a whole (e.g. over quota):
{
  code: "invalidProject",
  message: short string message for debugging
  errors: [ array of zero or more string messages for the user ]
}
On unexpected error (bug):
{
  code: "error",
  message: "Unexpected Error"
}
```

3 Test Data

You can use this project as one of your servermock projects. I've added an attachment and a file in a subfolder to make it a bit more interesting.