Mystatic_Sites Project Description

Overview

This project uses a site generator called pelican to generate static websites. At this writing it manages the content and generation for two sites.

1. www.bernatchez.net

   Hosted irn an AWS S3 bucket: www.bernatchez.net

2. www.ogopogo.biz

   Hosted in an AWS S3 bucket: www.ogopogo.biz

For either one, you go to the directory (www.bernatchez.net or www.ogopogo.biz), then modify .rst files under content/ or the pelicanconfig.py configuration file or the Makefile.

Then use make to rebuild.

```
make html
```

then use 'make html' to rebuild, and use 'make s3_upload' to upload to the site.

The credentials needed to access the site buckets must exist and be correct in ${HOME}/.s3cfg

For obvious security reasons we don't include those credentials in our repository.

To create or modify ${HOME}/.s3cfg you need to know the following information:

access_key = dummy secret_key = dummy gpg_passphrase = dummy bucket_location = us-east-1

anything else just go with the default values you are given.

```
s3cmd --configure
```

Upload the site.

```
make s3_upload
```

Modify again to put dummy in the credentials so that the credentials don't persist in the clear.

```
s3cmd --configure -c s3setup/s3cfg
```

Repository

The original central copy of this project's repository is:

git@github.com:pierrebernatchez/mypelican_static_sites.git

For more info see the notes:

buildnote orgnote

Notes about organizing mp3 files and tarballs for including them in web pages

We are maintaining the static web site www.bernatchez.net. The content is only proof of concept at this stage. It is mostly generated from the contents of the directory "www.bernatchez.net/content/compilations"

The compilations directory not intended to be maintained by git either, but we've incorporated a dozen sane mp3 files into the git repo to use as a test case.

tools:

compile_rst

generates rst format pages linking to the audio files in compilations. Also generates an include file which is used to insert a list of links pointing to raw files (currently tarballs) into a static web page. compile_rst has been incorporated into the Makefile's in the 'www....' subdirectories.

exfalso

gui for inspecting and editing tags. We use this to massage our raw audio content into something sane enough to be admitted to our web page compilations directory (currently ./www.bernatchez.net/content/compilations). Once we've done that we can regenerate our static web pages and upload them to s3 (using 'make html' and 'make s3_upload' in the www.bernatchez.net subdirectory)

I use it to fixup audio files. When I get a cd full of mp3 # files I give them all the same album name, and the same date # (a year like 2015). I try to arrange them in the same order # as appears on the cd label and give them sequential track # numbers starting from 1. I label them all 'artist' # 'unknown', and give them all the same genre (i.e. latin). # Then I go through them and try to replace the title and # artist with something useful. Then I rename all the files # to names like YYYY-albumname-track.mp3

id3info - Display id3 tag information.

id3cp - Copies tags from one file to another.

id3convert - Converts between id3v1 and id3v2 tags of an mp3 file.

id3tag - Tags an mp3 file with id3v1 and/or id3v2 tags.

mid3iconv - convert ID3 tag encodings: better replacement for above supposedly

mid3v2 - audio tag editor similar to 'id3v2'

mp3rename - Rename mp3 files based on id3tags

mp3tag - view and manipulate ID3v1 tags.

To get a list of genres: 'mp3tag -g list'

python-mutagen - lib I used to implement 'compile_rst'

Beware

If we were to actually maintain music track compilations on this static site, other than a proof of concept test as we have now, we would need put to gether better tools for adding tracks, editing tags and so on, and probably need to integrate that better with the compile script.