

Introduction: How to Use the bib2academic Package?

Peter Baumgartner

2018-08-04

bib2academic

bib2academic: Converting Bibtex Records to .md Files for the Hugo Academic Theme

bib2academic takes a .bib file and generates for each bib record a .md file. The name of the .md file is generated from the year and entry field. It has the format `year-month-day_key.md`.

Requirements:

- You have already installed a blogdown project with the Hugo Academic theme. If you do not know what this means read the Blogdown book. In the following text I assume that you are comfortable with the Hugo Academic theme and the blogdown environment.
- You have a valid bibtex file with your bibliography, e.g. it contains at least one full bibtex record. This file can have as many record as you want.

Preparations:

- You need to put you .bib file on the top level of your project directory or to figure out the path.

This conversion utility does not use the folders where the new files will finally live, e.g. `content/publication/` resp. `'static/files/citations/`. It is easier to recover you work if something goes wrong in the conversion process. For instance: You don't have to search the new generated files in your working folders, you just can empty or delete newly generated folders.

Call the bib2acad function:

The minimum call is: `bib2acad("<your/path/to/bibfilename>.bib")`.

This generates two folders in your working directory with the names `my-md-folder` and `my-bib-folder`. In these two folder will the generated new .md and .bib files collected.

The maximum call is: `bib2acad("<your/path/to/bibfilename>.bib", copybib = TRUE, abstract = TRUE, overwrite = TRUE)`.

This generates the .md files with abstracts, generates also .bib files and does overwrite files already generated in both folders (.md and .bib folder). This is not dangerous when you are working with folders especially created for this conversion process. To have more control you can change these parameters. Default values are:

- `copybib = TRUE`
- `abstract = TRUE`
- `overwrite = FALSE`

For many records in your .bib file the conversion procedure may take some time. I have included a progressbar so that you can see if the program is working. For my 190 bibtex records the procedure needed about 45 seconds on my MacBook Pro, 2,9 GHz Intel, 8GB RAM.

Install the conversion results:

If everything went smoothly make a copy of your `static/files/citations/` and `content/publication` folder. Then copy your `.bib` files into `static/files/citations/` and your `.md` files into `content/publication`.

To test if everything is converted correctly you must rebuild the website. Call the menu “Build -> More... -> Clean All” (or CTRL/CMD-SHIFT-B). Wait until the website is cleaned then call `blogdown::serve_site()` or - if you have RStudio Addins installed: CTRL-S.

Now your new publication files should be available on your website.

Trouble shooting:

If you can't see anymore your website but the directory index then something went wrong in the conversion procedure. The problem is that there is no error message in `blogdown`. You have to change to the terminal and call

```
hugo -v
```

directly.

Then you will get a more detailed report about. It maybe the case that you will get an error message similar like the following:

```
ERROR 2018/08/03 10:45:45 failed to parse page metadata for
publication/2018-01-01_baumgartner_dat literacy_2018.md":
Near line 5 (last key parsed 'publication'): invalid escape character '&';
only the following escape characters are allowed:
\b, \t, \n, \f, \r, \", \\, \uXXXX, and \UXXXXXXXX
for publication/2018-01-01_baumgartner_dat literacy_2018.md
```

This means that `hugo` has found in the `.md` file an invalid character. In this case it was the ampersand. Go to the file and look for the specified line (in my case it is line 5). Then you will see something like `\<special character>` (in my case: `\&`). Add another backlash. e.g. changing to `\\<special character>` (in my case: `\\&`).

The above example will now work out because I have added some error handling routines. But still there is a chance that other invalid characters throw errors. My suggestion:

1. Try to find and fix the problem in the file. Check it if everything is ok now with rebuilding your website followed by `blogdown::serve_site()`.
2. If you can't find the error most of the time problems appear in the `abstract` field of one of your `.md` files. Try to delete this field (or repeat the conversion procedure with `abstract = FALSE`). Again: Check if everything is ok now with rebuilding your website followed by `blogdown::serve_site()`.
3. If the error still persists try to delete the file(s) you are suspecting to throw the error. And again: check if this was the problem. This last procedure should help you to limit the problems to one or only some files.
4. If you could detect the error please give me feedback so that I could include an error handling routine for these cases.
5. If you cannot find the error please open an issue on my GitHub site and send me the `.bib` records where you assume that the problems were originated.