The problem

I choose to use the hugo-academic theme due to its good looks, simplicity, and “focus” towards researchers.

One nice feature of hugo-academic is that it includes out-of-the-box a “Publications” section, allowing researchers to easily create a list of their publication as a section of the website.

**Unfortunately**, in order to populate that list, users have to manually create one different .md file for each publication, by cutting and pasting several different info (e.g., title, authors, etc.) in a “simple” template

**THIS IS BORING!**

Since I was not in the mood of doing that, and no automatic solutions could be found (well, there appears to be a python one, but we are speaking R, here…), I decided to try and develop some script to automatically create the required md files starting from a BibTex list of my publications. Here are the results of that effort.

A possible solution

Preparing the BibTex file

To automatically create the publications md files, all you need is a (properly formatted) BibTex file. Since I did not have one ready, I created mine by exporting my publications list from [Scopus](https://www.scopus.com/authid/detail.uri?authorId=23003461400), but you could use any valid BibTeX file.

**One important thing, though, is that you have to be sure that the file is saved with UTF-8 encoding**. If you are not sure, you can open it in RStudio (or any decent text editor), and then re-save it with a forced encoding (in RStudio, you can use File-->Save with Encoding)

Creating an import script

Now, you need a script that reads the BibTex entries and use the data to populate one different mdfile for each publication. Below you can find my attempt at this.

bibtex\_2academic <- function(bibfile,  
 outfold,  
 abstract = FALSE,   
 overwrite = FALSE) {  
   
 require(RefManageR)  
 require(dplyr)  
 require(stringr)  
 require(anytime)  
   
 *# Import the bibtex file and convert to data.frame*  
 mypubs <- ReadBib(bibfile, check = "warn", .Encoding = "UTF-8") %>%  
 as.data.frame()  
   
 *# assign "categories" to the different types of publications*  
 mypubs <- mypubs %>%  
 dplyr::mutate(  
 pubtype = dplyr::case\_when(document\_type == "Article" ~ "2",  
 document\_type == "Article in Press" ~ "2",  
 document\_type == "InProceedings" ~ "1",  
 document\_type == "Proceedings" ~ "1",  
 document\_type == "Conference" ~ "1",  
 document\_type == "Conference Paper" ~ "1",  
 document\_type == "MastersThesis" ~ "3",  
 document\_type == "PhdThesis" ~ "3",  
 document\_type == "Manual" ~ "4",  
 document\_type == "TechReport" ~ "4",  
 document\_type == "Book" ~ "5",  
 document\_type == "InCollection" ~ "6",  
 document\_type == "InBook" ~ "6",  
 document\_type == "Misc" ~ "0",  
 TRUE ~ "0"))  
   
 *# create a function which populates the md template based on the info*  
 *# about a publication*  
 create\_md <- function(x) {  
   
 *# define a date and create filename by appending date and start of title*  
 if (!is.na(x[["year"]])) {  
 x[["date"]] <- paste0(x[["year"]], "-01-01")  
 } else {  
 x[["date"]] <- "2999-01-01"  
 }  
   
 filename <- paste(x[["date"]], x[["title"]] %>%  
 str\_replace\_all(fixed(" "), "\_") %>%  
 str\_remove\_all(fixed(":")) %>%  
 str\_sub(1, 20) %>%  
 paste0(".md"), sep = "\_")  
 *# start writing*  
 if (!file.exists(file.path(outfold, filename)) | overwrite) {  
 fileConn <- file.path(outfold, filename)  
 write("+++", fileConn)  
   
 *# Title and date*  
 write(paste0("title = \"", x[["title"]], "\""), fileConn, append = T)  
 write(paste0("date = \"", anydate(x[["date"]]), "\""), fileConn, append = T)  
   
 *# Authors. Comma separated list, e.g. `["Bob Smith", "David Jones"]`.*  
 auth\_hugo <- str\_replace\_all(x["author"], " and ", "\", \"")  
 auth\_hugo <- stringi::stri\_trans\_general(auth\_hugo, "latin-ascii")  
 write(paste0("authors = [\"", auth\_hugo,"\"]"), fileConn, append = T)  
   
 *# Publication type. Legend:*  
 *# 0 = Uncategorized, 1 = Conference paper, 2 = Journal article*  
 *# 3 = Manuscript, 4 = Report, 5 = Book, 6 = Book section*  
 write(paste0("publication\_types = [\"", x[["pubtype"]],"\"]"),   
 fileConn, append = T)  
   
 *# Publication details: journal, volume, issue, page numbers and doi link*  
 publication <- x[["journal"]]  
 if (!is.na(x[["volume"]])) publication <- paste0(publication,   
 ", (", x[["volume"]], ")")  
 if (!is.na(x[["number"]])) publication <- paste0(publication,  
 ", ", x[["number"]])  
 if (!is.na(x[["pages"]])) publication <- paste0(publication,  
 ", \_pp. ", x[["pages"]], "\_")  
 if (!is.na(x[["doi"]])) publication <- paste0(publication,  
 ", ", paste0("https://doi.org/",   
 x[["doi"]]))  
   
 write(paste0("publication = \"", publication,"\""), fileConn, append = T)  
 write(paste0("publication\_short = \"", publication,"\""),fileConn, append = T)  
   
 *# Abstract and optional shortened version.*  
 if (abstract) {  
 write(paste0("abstract = \"", x[["abstract"]],"\""), fileConn, append = T)  
 } else {  
 write("abstract = \"\"", fileConn, append = T)  
 }  
 write(paste0("abstract\_short = \"","\""), fileConn, append = T)  
   
 *# other possible fields are kept empty. They can be customized later by*   
 *# editing the created md*  
   
 write("image\_preview = \"\"", fileConn, append = T)  
 write("selected = false", fileConn, append = T)  
 write("projects = []", fileConn, append = T)  
 write("tags = []", fileConn, append = T)  
 *#links*  
 write("url\_pdf = \"\"", fileConn, append = T)  
 write("url\_preprint = \"\"", fileConn, append = T)  
 write("url\_code = \"\"", fileConn, append = T)  
 write("url\_dataset = \"\"", fileConn, append = T)  
 write("url\_project = \"\"", fileConn, append = T)  
 write("url\_slides = \"\"", fileConn, append = T)  
 write("url\_video = \"\"", fileConn, append = T)  
 write("url\_poster = \"\"", fileConn, append = T)  
 write("url\_source = \"\"", fileConn, append = T)  
 *#other stuff*  
 write("math = true", fileConn, append = T)  
 write("highlight = true", fileConn, append = T)  
 *# Featured image*  
 write("[header]", fileConn, append = T)  
 write("image = \"\"", fileConn, append = T)  
 write("caption = \"\"", fileConn, append = T)  
   
 write("+++", fileConn, append = T)  
 }  
 }  
 *# apply the "create\_md" function over the publications list to generate*  
 *# the different "md" files.*  
   
 apply(mypubs, FUN = function(x) create\_md(x), MARGIN = 1)  
}

Nothing fancy, here: I just use the [RefManageR](https://cran.r-project.org/web/packages/RefManageR/index.html" \t "_blank) package to read the BibTeX file, and then cycle over publications to create files properly formatted for hugo-academic use.

Running the script

All is left is to run the script:

my\_bibfile <- "/path/to/mybibtex.bib"  
out\_fold <- "/path/to/myoutfolder"  
  
bibtex\_2academic(bibffile = my\_bibfile,   
 outfold = out\_fold,   
 abstract = FALSE)

* The outfold argument allows specifying where the generated md files will be saved. Though in the end they will have to be moved to folder content/publication you may want to save them at first in a different folder to be able to check them before trying to deploy.
* The abstract argument specifies whether to include the abstract in the md or not.

Running the script will give you files like this one:

+++  
title = "Estimating canopy water content of poplar plantation from MIVIS data"  
date = "2006-01-01"  
authors = ["R. Colombo", "L. Busetto", "A. Marchesi", "M. Meroni", "C. Giardino"]  
publication\_types = ["1"]  
publication = "AIP Conference Proceedings, (852), \_pp. 242-249\_, https://doi.org/10.1063/1.2349350"  
publication\_short = ""  
abstract\_short = ""  
image\_preview = ""  
selected = false  
projects = []  
tags = []  
url\_pdf = ""  
url\_preprint = ""  
url\_code = ""  
url\_dataset = ""  
url\_project = ""  
url\_slides = ""  
url\_video = ""  
url\_poster = ""  
url\_source = ""  
math = false  
highlight = true  
[header]  
image = ""  
caption = ""  
+++

, where I tweaked a bit the hugo-academic format to include bibliographic info such as volume, number, pages and doi link. The files can then be further customized to include, for example, links to pdfs, images, etcetera.

After moving all the md files to content/publication, the publications section of your hugo-academicsite will be auto-populated, and should look more or less like this:



You can have a look at the final results on my (under construction) website [here](https://lbusett.netlify.com/publication/). I think it’s quite nice!

Final Notes

* My importing script is quite “quick and dirty”. It **does not attempt to deal with special characters, and even substitutes accented letters with “bare” letters to avoid rendering problems**. If someone more knowledgeable about encoding issues wants to try and improve it
* It may happen that your site will stop rendering when you put the new md files in content/publication. If so, the reason is probably that you have some “strange” formatting in some of the files. Usual suspects would be unproperly recognized **accents**, **math formulas** or other special characters in the **authors** and **abstract** fields. You will have to look into each file and remove any offending areas. (It happened to me a lot before properly saving to UTF-8)

script for importing publications from a "bibtex" file to a hugo-academic website

|  |
| --- |
| bibtex\_2academic <- function(bibfile, |
|  | outfold, |
|  | abstract = FALSE, |
|  | overwrite = FALSE) { |
|  |  |
|  | require(RefManageR) |
|  | require(dplyr) |
|  | require(stringr) |
|  | require(anytime) |
|  |  |
|  | # Import the bibtex file and convert to data.frame |
|  | mypubs <- ReadBib(bibfile, check = "warn", .Encoding = "UTF-8") %>% |
|  | as.data.frame() |
|  |  |
|  | # assign "categories" to the different types of publications |
|  | mypubs <- mypubs %>% |
|  | dplyr::mutate( |
|  | pubtype = dplyr::case\_when(document\_type == "Article" ~ "2", |
|  | document\_type == "Article in Press" ~ "2", |
|  | document\_type == "InProceedings" ~ "1", |
|  | document\_type == "Proceedings" ~ "1", |
|  | document\_type == "Conference" ~ "1", |
|  | document\_type == "Conference Paper" ~ "1", |
|  | document\_type == "MastersThesis" ~ "3", |
|  | document\_type == "PhdThesis" ~ "3", |
|  | document\_type == "Manual" ~ "4", |
|  | document\_type == "TechReport" ~ "4", |
|  | document\_type == "Book" ~ "5", |
|  | document\_type == "InCollection" ~ "6", |
|  | document\_type == "InBook" ~ "6", |
|  | document\_type == "Misc" ~ "0", |
|  | TRUE ~ "0")) |
|  |  |
|  | # create a function which populates the md template based on the info |
|  | # about a publication |
|  | create\_md <- function(x) { |
|  |  |
|  | # define a date and create filename by appending date and start of title |
|  | if (!is.na(x[["year"]])) { |
|  | x[["date"]] <- paste0(x[["year"]], "-01-01") |
|  | } else { |
|  | x[["date"]] <- "2999-01-01" |
|  | } |
|  |  |
|  | filename <- paste(x[["date"]], x[["title"]] %>% |
|  | str\_replace\_all(fixed(" "), "\_") %>% |
|  | str\_remove\_all(fixed(":")) %>% |
|  | str\_sub(1, 20) %>% |
|  | paste0(".md"), sep = "\_") |
|  | # start writing |
|  | if (!file.exists(file.path(outfold, filename)) | overwrite) { |
|  | fileConn <- file.path(outfold, filename) |
|  | write("+++", fileConn) |
|  |  |
|  | # Title and date |
|  | write(paste0("title = \"", x[["title"]], "\""), fileConn, append = T) |
|  | write(paste0("date = \"", anydate(x[["date"]]), "\""), fileConn, append = T) |
|  |  |
|  | # Authors. Comma separated list, e.g. `["Bob Smith", "David Jones"]`. |
|  | auth\_hugo <- str\_replace\_all(x["author"], " and ", "\", \"") |
|  | auth\_hugo <- stringi::stri\_trans\_general(auth\_hugo, "latin-ascii") |
|  | write(paste0("authors = [\"", auth\_hugo,"\"]"), fileConn, append = T) |
|  |  |
|  | # Publication type. Legend: |
|  | # 0 = Uncategorized, 1 = Conference paper, 2 = Journal article |
|  | # 3 = Manuscript, 4 = Report, 5 = Book, 6 = Book section |
|  | write(paste0("publication\_types = [\"", x[["pubtype"]],"\"]"), |
|  | fileConn, append = T) |
|  |  |
|  | # Publication details: journal, volume, issue, page numbers and doi link |
|  | publication <- x[["journal"]] |
|  | if (!is.na(x[["volume"]])) publication <- paste0(publication, |
|  | ", (", x[["volume"]], ")") |
|  | if (!is.na(x[["number"]])) publication <- paste0(publication, |
|  | ", ", x[["number"]]) |
|  | if (!is.na(x[["pages"]])) publication <- paste0(publication, |
|  | ", \_pp. ", x[["pages"]], "\_") |
|  | if (!is.na(x[["doi"]])) publication <- paste0(publication, |
|  | ", ", paste0("https://doi.org/", |
|  | x[["doi"]])) |
|  |  |
|  | write(paste0("publication = \"", publication,"\""), fileConn, append = T) |
|  | write(paste0("publication\_short = \"", publication,"\""),fileConn, append = T) |
|  |  |
|  | # Abstract and optional shortened version. |
|  | if (abstract) { |
|  | write(paste0("abstract = \"", x[["abstract"]],"\""), fileConn, append = T) |
|  | } else { |
|  | write("abstract = \"\"", fileConn, append = T) |
|  | } |
|  | write(paste0("abstract\_short = \"","\""), fileConn, append = T) |
|  |  |
|  | # other possible fields are kept empty. They can be customized later by |
|  | # editing the created md |
|  |  |
|  | write("image\_preview = \"\"", fileConn, append = T) |
|  | write("selected = false", fileConn, append = T) |
|  | write("projects = []", fileConn, append = T) |
|  | write("tags = []", fileConn, append = T) |
|  | #links |
|  | write("url\_pdf = \"\"", fileConn, append = T) |
|  | write("url\_preprint = \"\"", fileConn, append = T) |
|  | write("url\_code = \"\"", fileConn, append = T) |
|  | write("url\_dataset = \"\"", fileConn, append = T) |
|  | write("url\_project = \"\"", fileConn, append = T) |
|  | write("url\_slides = \"\"", fileConn, append = T) |
|  | write("url\_video = \"\"", fileConn, append = T) |
|  | write("url\_poster = \"\"", fileConn, append = T) |
|  | write("url\_source = \"\"", fileConn, append = T) |
|  | #other stuff |
|  | write("math = true", fileConn, append = T) |
|  | write("highlight = true", fileConn, append = T) |
|  | # Featured image |
|  | write("[header]", fileConn, append = T) |
|  | write("image = \"\"", fileConn, append = T) |
|  | write("caption = \"\"", fileConn, append = T) |
|  |  |
|  | write("+++", fileConn, append = T) |
|  | } |
|  | } |
|  | # apply the "create\_md" function over the publications list to generate |
|  | # the different "md" files. |
|  |  |
|  | apply(mypubs, FUN = function(x) create\_md(x), MARGIN = 1) |
|  | } |