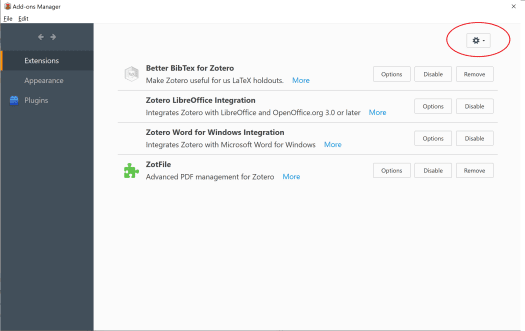
This blog is intended for researchers, PhD students, MD students and any other students who wish to have a robust and effective reference management setup. The blog has a particular focus on those using R markdown, Bookdown or LaTeX. Parts of the blog can also help setup Zotero for use with Microsoft Word. The blog has been designed to help achieve the following goals:

* Effective citation storage
  + Fast and easy **citation storage** (one-click from Chrome)
  + Fast and easy **PDF storage** using cloud storage
  + Immediate, automatic and standardised **PDF renaming**
  + Immediate, automatic and standardised **citation key** generation
* Effective citation integration with markdown etc.
  + Generation of citation keys which work with LaTeX and md (no non-standard characters)
  + Ability to **lock citation keys** so that they don’t update with Zotero updates
  + Storage of immediately updated .bib files for use with Rmd, Bookdown and LaTeX
  + Automated update of the .bib file in RStudio server

**Downloads and Setup**

For my current reference management setup I need the following software:

* [Zotero](https://www.zotero.org/download/)
  + **Zotero** comes with 300MB of free storage which allows well over 1000 references to be stored as long as PDFs are stored separately
  + From the same download page download the Chrome connector to enable the *“save to zotero”* function in Google Chrome
* [ZotFile](http://zotfile.com/)
  + **Zotfile** is a [Zotero plugin](https://www.zotero.org/support/plugins) which helps with PDF management, download the .xpi file and then open Zotero, go to “Tools → Add-Ons” and click the little cog in the top right corner and navigate to file to install (Figure 1)
* [Better BibTeX](https://github.com/retorquere/zotero-better-bibtex/releases/tag/v5.1.76)
  + **Better BibTeX** is a plugin to help generate **citation keys** which will be essential for writing articles in LaTeX, R Markdown or Bookdown
  + If the link doesn’t work go to [github](https://github.com/retorquere/zotero-better-bibtex.git) and scroll down to the **ReadMe** to find a link to download the .xpi file
  + The same approach is then used to install the **Better BibTeX** plugin for zotero (“Tools → Add-Ons”)
* Figure 1: Zotero Plugin/Add-on Installation

After downloading **Zotero, ZotFile** and **Better BibTeX** create an account on [Zotero online](https://www.zotero.org/user/login/).

In addition to the Zotero downloads this guide will focus on an efficient setup for writing with R markdown or Bookdown and assumes that you have access to the following software / accounts:

* Dropbox / Google Drive / other cloud storage service which allows APIs
  + It will also be necessary for these to be accessible using Windows Explorer or Mac Finder (there are many guides online for syncing [Google Drive](https://www.howtogeek.com/228989/how-to-use-the-desktop-google-drive-app/) and [Dropbox](https://www.dropboxforum.com/t5/Installation-and-desktop-app/How-do-I-put-Dropbox-on-my-File-Explorer-in-Win-10/td-p/313031) so that they appear in file explorers)
* RStudio (this is not 100% essential but it is far harder to use Rmd without it)
  + Packages which will be required for this setup include rdrop2 (if using dropbox, other packages are available to convert this setup to Google Drive etc.), encryptr, bookdown or Rmarkdown, tinytex and a LaTeX installation (the Bookdown author recommends using tinytex which can be installed by the similarly named R package: tinytex::install\_tinytex())

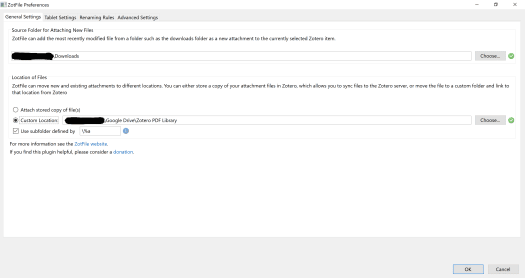
**Folder Setup**

When using Zotero it is a good ideal to create a folder in which you will store PDFs retrieved from articles. Ultimately it is optional whether or not PDFs are stored but if you have access to cloud storage with a good quota then it can make writing in Rmd etc. much faster as there is no requirement to search online for the original PDF. This folder should be set up in **Google Drive, Dropbox** or another cloud storage service which can be accessed from your own computer through the file explorer.

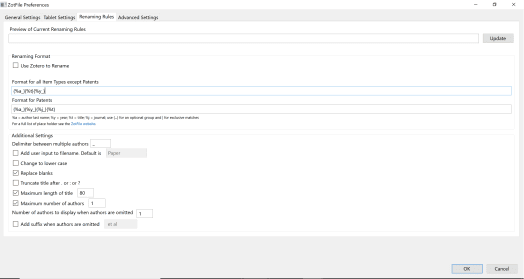
A second folder may be useful to store bibliographies which will be generated for specific projects or submissions. Again this folder should be made available in cloud storage.

**ZotFile Preferences**

To setup Zotero so that retrieved PDFs are automatically stored and renamed in the cloud storage without consuming the Zotero storage quota go to “Tools → ZotFile Preferences” and on the first tab: **General Settings** and set the folder and subfolder naming strategy for PDFs. I have set the location of the files to a Custom location and in this case used the path to a Google Drive folder (~\Google Drive\Zotero PDF Library). ZotFile will also store retrieved PDFs in subfolders to help with finding PDFs at a later date. The current setup I use is to create a subfolder with the first author surname so that all papers authored by one (or more) author with the same name are stored together using the \%a in the subfolder field (Figure 2). Other alternatives are to store PDFs in subfolders using year (\%y); journal or publisher (\%w); or item type (\%T).

* Figure 2: ZotFile Preferences

Next the **Renaming Rules** tab can be configured to provide sensible names to each of the files (this is essential if PDFs are not to be stored as random strings of characters which provide no meaning). In this tab I have set the format to: {%a\_}{%y\_}{%t} which provides names for the PDFs in the format of: Fairfield\_2019\_Gallstone\_Disease\_and\_the\_Risk\_of\_Cardiovascular\_Disease.pdf. I find that this shows author, year and first word of title without needing to expand the file name (Figure 3).

* Figure 3: ZotFile PDF Renaming Preferences

I have not changed any of the default settings in either the **Tablet Settings** or **Advanved Settings** tabs apart from removing special characters in the **Advanced Settings** (this stops things from breaking later).

**General Zotero Settings**

Zotero has several configurable settings (accessed through: “Edit → Preferences”) and I have either adopted the defaults or made changes as follows:

**General**:

* I have ticked the following:
  + Automatically attach associated PDFs
  + Automatically retrieve metadata for PDFs
  + Automatically rename attachments using parent metadata
  + Automatically tag items with keywords and subject headings
  + All options in *Group* section
* I have left the following unticked:
  + Automatically take snapshots
  + Rename linked files

**Sync**:

* Enter the account details
* Tick sync automatically
* **Untick** sync full text (if you choose to save PDFs then syncing full text will quickly consume the 300MB quota)

**Search**:

* Left unchanged

**Export**:

* Left unchanged

**Cite**:

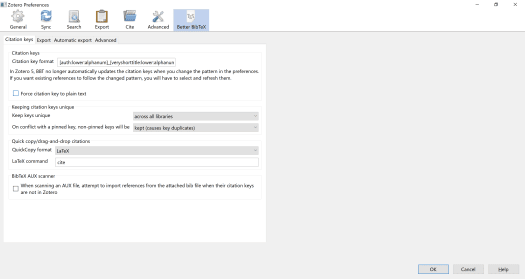
* There are several sensible defaults but if there is a new citation style you wish to be able to use in Microsoft Word for example then click “Get additional styles” as there is probably a version that you need already created. You can click the “+” button to add a style from a .csl file if you have one already. Finally, if you are desperate for a style that doesn’t already exist then you can select a citation style and click Style Editor and edit the raw .csl file.
* In the Word Processors subtab (on the main **Cite** tab), you can install the Microsoft Word add-in to allow Zotero to work in Microsoft Word.

**Advanced**:

* I changed nothing on the General subtab
* In the Files and Folders subtab I have selected the path to base directory for attachments
* I have not changed the Shortcuts subtab
* I have not changed the Feeds subtab

**Better BibTex**:

* In this section I have set my **Citation Key** format to [auth:lower:alphanum]\_[year:alphanum]\_[veryshorttitle:lower:alphanum]\_[journal:lower:clean:alphanum] (Figure 4). This generates a citation key for each reference in the format of fairfield\_2019\_gallstones\_scientificreports or harrison\_2012\_hospital\_bmj. It always takes the first author’s surname, the year, the first word of the title and the journal abbreviation if known. The clean and alphanum arguments to this field are used to remove unwanted punctuation which can cause citation to fail in LaTeX.

Figure 4: Better BibTeX Citation Key

Once the settings have been configured if you already had references stored in Zotero and wish to change the citation key for old references select your entire library root (above all folders), select all references, right click and use “Better BibTex → Refresh BibTeX Key” and all of the citation keys should be updated.

**Creating a .bib file**

For referencing in a new project, publication or submission it may be helpful to have a dynamic .bib file that updates with every new publication and can be accessed from any device through cloud storage.

To set up a .bib file, first find the folder that you wish to create the file from (this should be the folder which contains any citations you will use and ideally not the full library to cut down on unnecessary storage and syncing requirements). Note that the .bib file will generate a bibliography from any citations stored *directly* in the folder when using default settings. This prevents use of subfolders which I find particularly helpful for organising citations and I have therefore changed the setting so that folders also show any citations stored in subfolders. To make this change go to “Edit Preferences” and select the “Advanced” tab and at the bottom of the “General” subtab select “Config Editor”. This will bring up a searchable list of configurations (it may show a warning message before this) and search in the search box for “extensions.zotero.recursiveCollections”. Set “Value” to TRUE and then when you click a folder you should see all of the citations also stored in subfolders.

Right click the folder and select “Export Collection”. A pop-up window will appear at which point select “Keep Updated” and if using RStudio desktop save the file in the directory where you have your Rmd project files. If you are working with RStudio server then save the file in a cloud storage location which will then be accessed from the server. I have a .bib file stored in Dropbox which I access from RStudio server.

**Linking Dropbox and RStudio Server to Access the .bib File**

The following covers linking Dropbox to RStudio server but could be adapted to cover another cloud storage service.

Dropbox provides a token to allow communication between different apps. The rdrop2 package is what I used to create a token to allow this. I actually created the token on RStudio desktop as I couldn’t get the creation to work on the server but this is perfectly ok.

**Caution**: The token generated by this process could be used to access your Dropbox from anywhere using RStudio if you do not keep it secure. If somebody were to access an unencrypted token then it would be equivalent to handing out your email and password. I therefore used the encryptr package to allow safe storage of this token.

**Token Creation**

Open Rstudio desktop and enter the following code:

library(rdrop2)

library(encryptr)

# Create token

token <- drop\_auth()

# Save token

saveRDS(token, "droptoken.rds")

# Encrypt token

genkeys() # Default file names are id\_rsa and id\_rsa.pub

encrypt\_file("droptoken.rds", "droptoken.rds.encryptr.bin")

encrypt\_file(".httr-oauth", ".httr-oauth.encryptr.bin")

# Same details should appear later

drop\_acc()

# Remove token from local environment

rm(token)

# Delete the unencrypted files

system("rm droptoken.rds")

system("rm .httr-oauth")

The code will create two files, a token and the .httr-oauth file from which a token can also be made. The encryptr package can then encrypt the files using a public / private key pair. It is essential that the password that is set when using genkeys() is remembered otherwise the token cannot then be used. In this case the original token can’t be retrieved but could be created again from scratch.

The following files will then be needed to upload to the RStudio server:

* droptoken.rds.encryptr.bin – or the name provided for the encrypted Dropbox token
* id\_rsa – or the name provided for the private key from the private / public key pair

**Dropbox Linkage for Referencing the .bib File**

Now that the encrypted token and necessary (password-protected) private key are available in RStudio server, the following can be saved as a separate script. The script is designed to read in and decrypt the encrypted token (this will require a password and should be done if the .bib file needs updated). Only the drop\_download() needs repeated if using the token again during the same session. The token should be cleared at the end of every session for additional security.

library(rdrop2)

library(encryptr)

# \*\*\*\*\*\*\*\* WARNING \*\*\*\*\*\*\*\*

# Losing the unencrypted token will give anyone

# complete control of your Dropbox account

# If you are concerned this has happened,

# you can then revoke the rdrop2 app from your

# dropbox account and start over.

# \*\*\*\*\*\*\*\* WARNING \*\*\*\*\*\*\*\*

safely\_extract\_dropbox\_token <- function(encrypted\_db\_token = NULL, private\_key\_file = NULL){

decrypt\_file(encrypted\_db\_token, file\_name = "temporary\_dropbox\_token.rds", private\_key\_path = private\_key\_file)

token <<- readRDS("temporary\_dropbox\_token.rds")

system("rm temporary\_dropbox\_token.rds")

}

safely\_extract\_dropbox\_token(encrypted\_db\_token = "droptoken.rds.encryptr.bin", private\_key\_file = "id\_rsa")

# Then pass the token to each drop\_ function

drop\_acc(dtoken = token)

# The path is the Dropbox file location

drop\_download(path = "My\_Dropbox\_Home\_Directory/Zotero Library/my.bib",

local\_path = "my.bib",

dtoken = token,

overwrite = TRUE)

Now that the .bib file has been created and is stored as “my.bib” in the local directory, it should update whenever the token is loaded and drop\_download() is run.

**Final Result**

On clicking “Save to Zotero” button in Chrome and running drop\_download() the following should all happen almost instantaneously:

* Zotero stores a new reference
* A PDF is stored in the cloud storage having been named appropriately
* A link to the PDF is stored in Zotero (without using up significant memory)
* A citation key is established for the reference in a standardised format without conflicts
* Pre-existing citation keys which have been referenced earlier in the writing of the paper are not altered
* A .bib file is updated in the RStudio server directory
* And much unwanted frustration of reference management is resolved

This is my current reference management system which I have so far found to be very effective. If there are ways you think it can be improved I would love to hear about them.