

Instantly share code, notes, and snippets.



dropmeaword / browser_history.md

Last active 3 weeks ago

<> Code - Revisions 38 ☆ Stars 135 🔗 Forks 30

Embed ▼

<script src="https://g.



Download ZIP

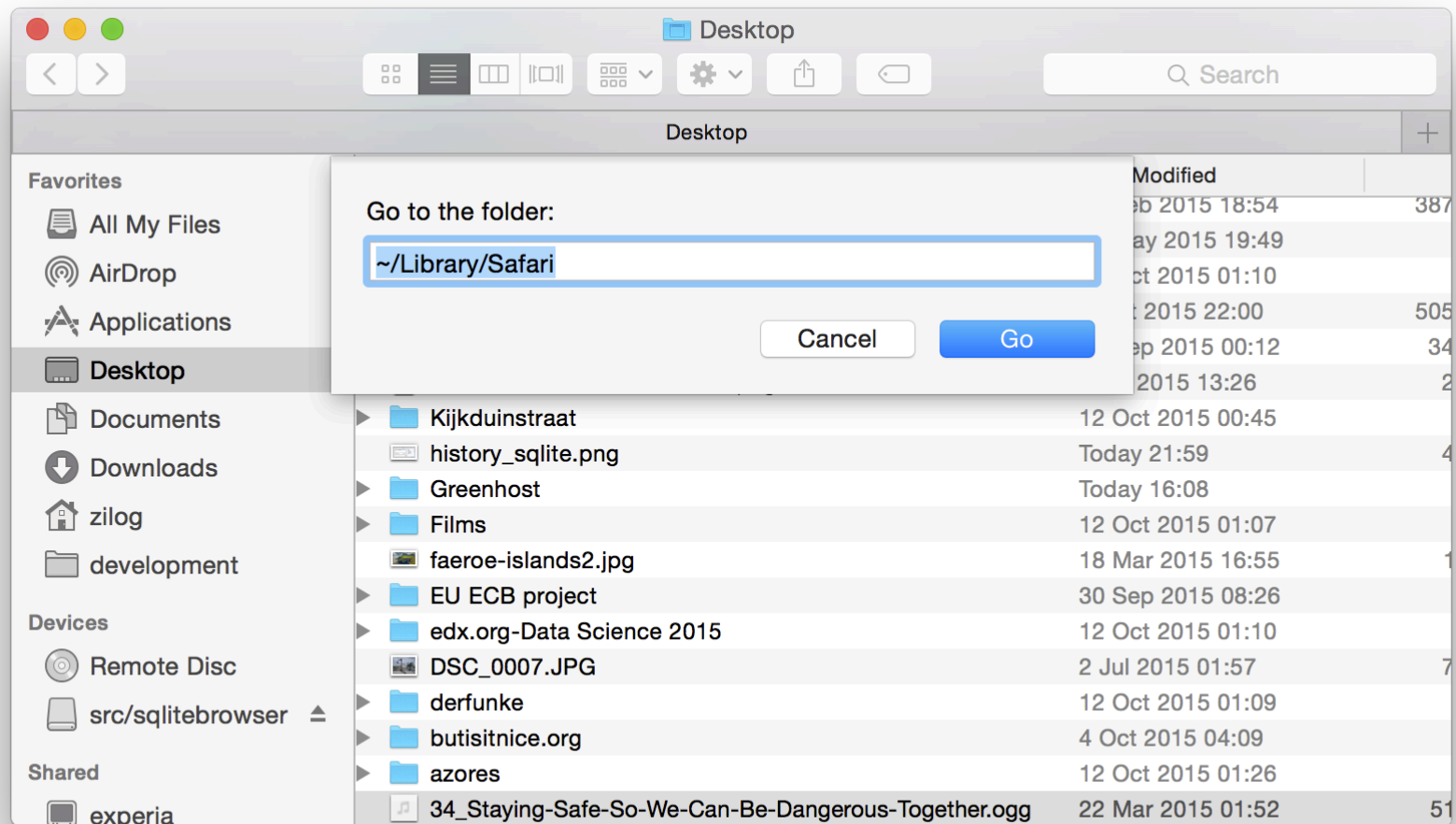
Playing around with Chrome's history

browser_history.md

Browser histories

Unless you are using Safari on OSX, most browsers will have some kind of free plugin that you can use to export the browser's history. So that's probably the easiest way. The harder way, which seems to be what Safari wants is a bit more hacky but it will also work for other browsers. Turns out that most of them, including Safari, have their history saved in some kind of [sqlite](#) database file somewhere in your home directory.

The OSX Finder cheats a little bit and doesn't show us all the files that actually exist on our drive. It tries to protect us from ourselves by hiding some system and application-specific files. You can work around this by either using the terminal (my preferred method) or by using the `Cmd + Shift + G` in Finder.



Once you locate the file containing the browser's history, copy it to make a backup just in case we screw up.

Safari

Safari is a little trickier than other browsers. AFAIK, Apple doesn't currently offer extensions to export data from its browser. So we have to get our hands dirty and hack it out of the browser.

But I found that there's another way to do it, takes a few steps but it's rather easy. Turns out that all browsers, Firefox, Chrome and Safari all keep their histories in a `sqlite` database. It's an embedded database system that is often found in apps. So this means that you can use any `sqlite` database viewer to open those files and inspect them. This one is pretty good and simple: <http://sqlitebrowser.org/>

1. Download the `sqliteBrowser` dmg for osx.
2. Install in your applications folder.
3. open `sqliteBrowser`
4. The file containing your Safari history is in `~/Library/Safari/History.db`
5. Open that file with `sqliteBrowser`, your OSX file open dialog probably will not show the `~/Library` directory, that's Apple trying to protect you from yourself. While the file open dialog is open, press `Cmd+Shift+G`, you will see the Go To Dialog. Paste this `~/Library/Safari/`, and you'll see the History DB file there.
6. Go to the *execute SQL* tab.
7. Paste this code:

```
SELECT
    datetime(hv.visit_time + 978307200, 'unixepoch', 'localtime') as last_visited,
    hi.url,
    hv.title
FROM
    history_visits hv,
    history_items hi
WHERE
    hv.history_item = hi.id;
```

8. This basically get's three fields from the database, the time of the last visit, the title and the url.
9. If everything goes well you should see something like this screenshot.
10. Use the button beneath the result set to export those results to CSV format.

The screenshot shows the DB Browser for SQLite application. The main window displays a SQL query in the 'SQL 1' editor:

```
1 SELECT datetime(hv.visit_time + 978307200, 'unixepoch', 'localtime') as last_visited, hv.title, hi.url
2 FROM history_visits hv, history_items hi WHERE hv.history_item = hi.id;
```

Below the query, the results are displayed in a table with 3 columns: last_visited, title, and url. The table contains 3 rows of data:

| | last_visited | title | url |
|-----|---------------------|-------------------------|-------------------------------|
| 130 | 2015-10-15 21:29:01 | Nicolas Collins: Look | http://www.nicolascollins.cor |
| 131 | 2015-10-15 21:29:10 | La Sebastiana | http://www.nicolascollins.cor |
| 132 | 2015-10-15 21:29:06 | Nicolas Collins: Photos | http://www.nicolascollins.cor |

Below the table, a status bar indicates: 132 Rows returned from: SELECT datetime(hv.visit_time + 978307200, 'unixepoch', 'localtime') as last_visited, hv.title, hi.url FROM history_visits hv, history_items hi WHERE hv.history_item = hi.id; (took 1ms)

The right sidebar shows the 'DB Schema' panel, listing the database structure:

- Tables (5): history_items, history_tombstones, history_visits, metadata, sqlite_sequence
- Indices (9): history_items__domain_expansion, history_tombstones__end_time, history_tombstones__generation, history_visits__last_visit, history_visits__origin, sqlite_autoindex_history_items_1, sqlite_autoindex_history_visits_1, sqlite_autoindex_history_visits_2, sqlite_autoindex_metadata_1
- Views (0)
- Triggers (0)

At the bottom of the main window, there are buttons for 'SQL Log', 'Plot', and 'DB Schema'. A tooltip 'Export to CSV Save as view' is visible over the 'DB Schema' button.

If this method fails for you, try these other approaches:

- [Here's a little guide.](#)
- You can also try this guy's Ruby script [this Ruby script](#).

Chrome

Chrome keeps its history, in a file that is typically located at `~/Library/Application Support/Google/Chrome/Default/History`. You can use the same method explained above to work with this file, but because the database structure is slightly different, we will to modify the sql code slightly. Here's a clip that will work with Chrome.

```
SELECT
  datetime(last_visit_time/1000000-11644473600, "unixepoch") as last_visited,
  url,
  title,
  visit_count
FROM urls;
```

Alternatively there are some Chrome add-ons that will export historical data. You can try <https://github.com/christiangenco/chrome-export-history> or [History Trends Unlimited](#)

Firefox

Same technique as above, this time the file is called `places.sqlite` and is located in the profile directory located here `~/Library/Application Support/Firefox/Profiles/`.

```
SELECT
  datetime(moz_historyvisits.visit_date/1000000, 'unixepoch'),
  moz_places.url,
  moz_places.title
FROM
  moz_places,
  moz_historyvisits
WHERE
  moz_places.id = moz_historyvisits.place_id;
```

Windows browsers

If you want to get the browser history from a windows browser there are some tools that can help you. Try [Browsing History View](#). Seems to support all common browsers. I haven't tested this because I don't use Windows.

Working with history data

Once you have exported the data, you should end up with a CSV (Comma Separated Values) file that looks a bit like this one.

```
2015-10-15 21:29:01,http://www.nicolascollins.com/look.htm,Nicolas Collins: Look
2015-10-15 21:29:10,http://www.nicolascollins.com/neruda.htm,La Sebastiana
2015-10-15 21:29:06,http://www.nicolascollins.com/photos.htm,Nicolas Collins: Photos
2015-03-25 14:25:53,"http://www.google.nl/url?
sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0CCcQFjAA&url=http%3A%2F%2Frefspace.com%2Fquotes%2FCarl_Sagan%2FQ263&ei=XLc
5gKAC&usg=AFQjCNFpaF390DuN1-cS6Y6ghDPWyXDvmQ&bvm=bv.89184060,d.ZWU&cad=rja",
2015-03-25 14:25:54,http://refspace.com/quotes/Carl_Sagan/Q263,Carl Sagan Quote: All civilizations become either
spacefaring or extinct. - Refspace
2015-03-25 14:25:56,http://www.dailygalaxy.com/my_weblog/2012/09/what-will-a-civilization-a-million-years-ahead-
of-earth-look-like-weekend-feature.html,"""What Will a Civilization a Million Years Ahead of Earth Look Like?""
2015-03-25 14:25:55,"http://www.google.nl/url?
sa=t&rct=j&q=&esrc=s&source=web&cd=4&ved=0CD4QFjAD&url=http%3A%2F%2Fwww.dailygalaxy.com%2Fmy_weblog%2F2012%2F09%2Fwh
will-a-civilization-a-million-years-ahead-of-earth-look-like-weekend-feature.html&ei=XLcSVZj2HcK7Pf-
5gKAC&usg=AFQjCNEfnUdBF6fjsHXlxChXJrrZmXYog&bvm=bv.89184060,d.ZWU&cad=rja",
```



Importing a fresh history to a clean Chrome

In order to analyze the data, we are going to import it into Chrome.

1. Copy the code bellow and save it to a file named `importhistory.py` in your desktop.
2. Save the CSV file you want to import as `history.csv` in your desktop.
3. Open Chrome and clean the browsing history data by going to the menu `Chrome > Clear Browsing Data...`. Your copy of chrome should now have a clean history.
4. Import our *hacked* history from another browser, into our fresh Chrome. Tadam! Your data is now ready for analysis.

```
#!/usr/bin/env python
## See: https://gist.github.com/dropmeaword/9372cbeb29e8390521c2
```

```
import csv
from os.path import expanduser
import sqlite3
import time, datetime

# the CSV data you obtained from somebody else's browser
CSV_FILE="~/Desktop/history.csv"
# the place where Chrome stores its history and that we will use
# to analyze our source data
HISTORY_DB='~/Library/Application Support/Google/Chrome/Default/History'

# open destination database
conn = sqlite3.connect(expanduser(HISTORY_DB))
# get a data cursor to work with the database
c = conn.cursor()

def chrome_list_history():
    """ list existing history in chrome """
    result = c.execute('SELECT datetime(last_visit_time/1000000-11644473600, "unixepoch") as last_visited,
last_visit_time, url , title, visit_count FROM urls;')

    for row in result:
        print row

def chrome_insert_visit(url, title, when=13089235804776906):
    """ there must be an entry in urls and another in visits for Chrome to recognize the history entry """
    c.execute('INSERT INTO urls (last_visit_time, url, title, visit_count) VALUES (?, ?, ?, ?)', (when, url,
title, 1))
    lid = c.lastrowid
    c.execute('INSERT INTO visits (url, visit_time, visit_duration, from_visit, transition, segment_id) VALUES
(?, ?, ?, ?, ?, ?)', (lid, when, 42047037, 0, 805306376, 0))

def import_to_chrome():
    print "Opening history CSV data from", CSV_FILE
    with open(expanduser(CSV_FILE), 'rb') as f:
        reader = csv.reader(f, delimiter=',', quotechar='')
```

```
for row in reader:
    try:
        when = time.mktime(datetime.datetime.strptime(row[0], "%Y-%m-%d %H:%M:%S").timetuple())
        when = int(when + 11644473600)*1000000
        timestamp = datetime.datetime.strptime(row[0], "%Y-%m-%d %H:%M:%S") #- datetime.datetime(1970,
1, 1)

        print "Importing: ", when, row
        chrome_insert_visit(unicode(row[1]), unicode(row[2]), when)
    except ValueError as e:
        pass

print
print "***70
print

import_to_chrome()
#chrome_list_history()
#chrome_insert_visit('http://www.kimchi.nl', 'Kimchi')

print "Closing DB"
conn.commit()
conn.close()
```

Trouble?

You might get some errors while running the python code. If the error looks like this:

```
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ImportError: No module named sqlite3
```

You are missing the library `sqlite3`, which allows our code to work with databases. Simply install it by typing this command in the command line:


```
$ sudo pip install sqlite3
```

Note that you do not need to type the dollar sign, this is just a convention used in documentation and tutorials to indicate that this is a command to be typed in the terminal.

If your OS complains that `pip` doesn't exist, you can install it by typing.

```
$ sudo easy_install pip
```

Some links for reference

- [Chrome Transition Values](#)
- [Chrome Transitions](#)
- [How Chrome Stores Browsing History](#)



Roman2K commented on Oct 1, 2016 • edited ▼

@dropmeaword Very useful and well written. Used it as a base to import Safari history into Chrome. I wrote a script for this based on the info found in your gist. You may be interested: [Roman2K/hist_safari2chrome](#)



PBXY commented on Sep 20, 2017

Thanks for this - **@dropmeaword**

I am working on a Mac and trying to export my Safari browser history to Firefox via Chrome

I have successfully created the `importhistory.py` file on my desktop (appears to be in Xcode) , and the `.history.csv` file (appears to be in Open Office) after following your very clear instructions. I also have a clean Chrome copy.

However, (and I am a novice in this field), I am probably being stupid but how do I get the history into Chrome?
(Once I get history into Chrome I know I can get Firefox to import easily from Chrome)

I would be grateful if you could spell out to me step by step how to do 'stage 4' - " Import our hacked history from another browser, into our fresh Chrome. Tadam! Your data is now ready for analysis. "

Thanks in advance, and thanks for your trouble in providing this hack.



proplayer99 commented on Dec 4, 2020

Really great written. I'm impressed.



Corginyan commented on Aug 31, 2024

Thank you very much for the tip!

I don't know the best place to post this, so here are some of my experiments with the Chrome history database, maybe they will be useful to someone. I don't have enough knowledge of SQL and Chrome infrastructure, so if you want to use this information, it will be **at your own risk**. And be sure to backup the history file.

Clearing website browsing history

Each change in the Google translate input form created a new entry in the history, and when I finally decided to fix it - there were already thousands of entries. I realized that editing history through the browser interface would be too slow. So I did this:
First, I deleted all the entries related to the link from the `visits` table:

```
DELETE FROM visits WHERE url IN (SELECT id FROM urls WHERE url LIKE 'https://translate.google.com%')
```

Then I deleted all the corresponding entries from the `urls` table:

```
DELETE FROM urls WHERE url LIKE 'https://translate.google.com%'
```

Viewing history became cleaner and works much faster. By the way, to prevent this from happening again, I installed the "History Blacklist" extension, maybe there are more decent ways.

Clear address bar history

Typing in the browser address bar became quite slow, so I thought that it might be somehow related to autocompletion. So I decided to delete all data from the `keyword_search_terms` table:

```
DELETE FROM keyword_search_terms
```

I lost the current autocompletion history, but typing in the address bar became an order of magnitude faster.

P.S.:

The size of the database file on the disk can be quite large, mine took up more than a gigabyte. After editing, the size of the database did not change, apparently the once allocated space is not freed when deleting entries. Most likely it can be compressed, but I have not studied this issue.