
sqlitebiter Documentation

Release 0.36.1

Tsuyoshi Hombashi

Mar 28, 2022

TABLE OF CONTENTS

| | | |
|----------|--|-----------|
| 1 | sqlitebiter | 1 |
| 1.1 | Summary | 1 |
| 1.2 | Features | 1 |
| 2 | Installation | 3 |
| 2.1 | Installation: pip (Python package manager) | 3 |
| 2.2 | Installation: apt | 3 |
| 2.3 | Installation: dpkg (.deb package) | 3 |
| 2.4 | Installation: Windows | 3 |
| 2.5 | Installation: Windows (PowerShell) | 4 |
| 2.6 | Installation: brew for macOS | 4 |
| 2.7 | Command Completion (bash/zsh) | 4 |
| 3 | Dependencies | 5 |
| 3.1 | Python package dependencies | 5 |
| 3.1.1 | Google Sheets dependencies (Optional) | 5 |
| 3.1.2 | Misc dependencies (Optional) | 5 |
| 3.2 | Dependencies other than Python packages (Optional) | 5 |
| 4 | Usage | 7 |
| 4.1 | sqlitebiter command help | 7 |
| 4.2 | Create a SQLite database from CSV/Excel/JSON/SQLite/etc. files | 8 |
| 4.2.1 | Examples | 8 |
| 4.2.2 | sqlitebiter file subcommand help | 9 |
| 4.2.3 | Supported data formats | 10 |
| 4.2.4 | Table naming convention | 11 |
| 4.3 | Create a SQLite database from URL | 11 |
| 4.3.1 | Example | 11 |
| 4.3.2 | sqlitebiter url subcommand help | 14 |
| 4.4 | Create a SQLite database from stdin | 15 |
| 4.4.1 | Examples | 15 |
| 4.4.2 | sqlitebiter stdin subcommand help | 15 |
| 4.5 | Create a SQLite database from Google Sheets | 15 |
| 4.5.1 | Requirements | 15 |
| 4.5.2 | Example | 16 |
| 4.5.3 | sqlitebiter gs subcommand help | 16 |
| 5 | Changelog | 17 |
| 6 | Sponsors | 19 |

| | | |
|----------|---------------------------|-----------|
| 7 | Indices and tables | 21 |
| 8 | Links | 23 |
| 9 | Indices and tables | 25 |

SQLITEBITER

1.1 Summary

sqlitebiter is a CLI tool to convert CSV / Excel / HTML / JSON / Jupyter Notebook / LDJSON / LTSV / Markdown / SQLite / SSV / TSV / Google-Sheets to a SQLite database file.

1.2 Features

- **Create a SQLite database file from:**
 - **File(s):**
 - * CSV / Tab separated values (TSV) / Space separated values (SSV)
 - * Microsoft Excel TM
 - * HTML
 - * JSON
 - * Jupyter Notebook
 - * Labeled Tab-separated Values (LTSV)
 - * Line-delimited JSON(LDJSON) / NDJSON / JSON Lines
 - * Markdown
 - * Mediawiki
 - * SQLite
 - Google Sheets
 - URL (scrape tabular data from web pages)
- Multi-byte character support

- Automatic file encoding detection

INSTALLATION

2.1 Installation: pip (Python package manager)

```
pip install sqlitebiter
```

2.2 Installation: apt

You can install the package by apt via a Personal Package Archive (PPA):

```
sudo add-apt-repository ppa:thombashi/ppa
sudo apt update
sudo apt install sqlitebiter
```

2.3 Installation: dpkg (.deb package)

The following commands will download the latest .deb package from the [release page](#), and install it.

```
curl -sSL https://raw.githubusercontent.com/thombashi/sqlitebiter/master/scripts/
↪installer.sh | sudo bash
```

2.4 Installation: Windows

sqlitebiter can be used in Windows environments without Python installation as follows:

1. Navigate to <https://github.com/thombashi/sqlitebiter/releases>
2. Download the latest version of the `sqlitebiter_win_x64.zip`
3. Unzip the file
4. Execute `sqlitebiter.exe` in either Command Prompt or PowerShell

2.5 Installation: Windows (PowerShell)

The following commands will download the latest execution binary from the [release page](#) to the current directory.

```
wget https://github.com/thombashi/sqlitebiter/raw/master/scripts/get-sqlitebiter.ps1 -  
-OutFile get-sqlitebiter.ps1  
Set-ExecutionPolicy Unrestricted -Scope Process -Force; .\get-sqlitebiter.ps1
```

2.6 Installation: brew for macOS

```
$ brew tap thombashi/sqlitebiter  
$ brew install sqlitebiter
```

- [Homebrew Formula](#)

2.7 Command Completion (bash/zsh)

```
setup command completion for bash:  
  
    sqlitebiter completion bash >> ~/.bashrc  
  
setup command completion for zsh:  
  
    sqlitebiter completion zsh >> ~/.zshrc
```


DEPENDENCIES

Python 3.6+

3.1 Python package dependencies

- Mandatory dependencies (automatically installed)

3.1.1 Google Sheets dependencies (Optional)

Extra Python packages are required to install to use Google Sheets feature (*gs* subcommand):

- `gspread`
- `oauth2client`
- `pyOpenSSL`

The extra packages can be installed with the following *pip* command;

```
$ pip install sqlitebiter[gs]
```

note: binary packages include these dependencies

3.1.2 Misc dependencies (Optional)

- `lxml`
- `py pandoc`
 - required when converting MediaWiki files

3.2 Dependencies other than Python packages (Optional)

- `libxml2` (faster HTML/Markdown conversion)
- `pandoc` (required when converting MediaWiki files)

4.1 sqlitebiter command help

sqlitebiter has following subcommands:

- **file:** Convert tabular data within CSV/Excel/HTML/JSON/LTSV/Markdown/SQLite/TSV file(s) to a SQLite database file.
 - *Create a SQLite database from CSV/Excel/JSON/SQLite/etc. files*
- **url:** Scrape tabular data from a URL and convert data to a SQLite database file.
 - *Create a SQLite database from URL*
- **gs:** Convert a spreadsheet in Google Sheets to a SQLite database file.
 - *Create a SQLite database from Google Sheets*
- **configure:** Configure the application settings

```
Usage: sqlitebiter [OPTIONS] COMMAND [ARGS]...
```

Options:

```
-o, --output-path PATH          Output path of the SQLite database file.
                                Defaults to 'out.sqlite'.
-a, --append                    Append table(s) to existing database.
--add-primary-key PRIMARY_KEY_NAME
                                Add 'PRIMARY KEY AUTOINCREMENT' column to a
                                converted table with the specified name.
--convert-config TEXT           [experimental] Configurations for data
                                conversion. The option can be used only for
                                url subcommand.
-i, --index INDEX_ATTR         Comma separated attribute names to create
                                indices.
--no-type-inference            All of the columns assume as TEXT data type
                                in creating tables.
--type-hint-header             Use headers suffix as type hints. If there
                                are type hints, converting columns by
                                datatype corresponding with type hints. The
                                following suffixes can be recognized as type
                                hints (case insensitive): "text": TEXT
                                datatype. "integer": INTEGER datatype.
                                "real": REAL datatype.
--matrix-formatting [header_aligned|trim]
```

(continues on next page)

(continued from previous page)

```

header_aligned: fitting table data to header
columns. trim: fitting table data to minimum
column size. Defaults to header_aligned.
--replace-symbol TEXT      Replace symbols in attributes.
-v, --verbose
--max-workers WORKERS     Specify maximum number of workers that the
command may use. defaults to 1.
--debug                    For debug print.
-q, --quiet                Suppress execution log messages.
-h, --help                Show this message and exit.

```

Commands:

```

completion  A helper command to setup command completion.
configure   Configure the following application settings:
file        Convert tabular data within CSV/Excel/HTML/JSON/Jupyter...
gs          Convert a spreadsheet in Google Sheets to a SQLite database...
stdin       Convert tabular data within CSV/HTML/JSON/Jupyter...
url         Scrape tabular data from a URL and convert data to a SQLite...
version     Show version information

```

4.2 Create a SQLite database from CSV/Excel/JSON/SQLite/etc. files

sqlitebiter file is a subcommand to convert tabular data file(s) to a SQLite database file.

4.2.1 Examples

Using wildcard to convert multiple files. File formats are automatically detected from the extensions.

Example

```

$ ls
sample_data.csv  sample_data.xlsx  sample_data_multi.json  sample_data_
↳single.json
$ sqlitebiter -o sample.sqlite file *
[INFO] sqlitebiter file: convert 'sample_data.csv' to 'sample_data' table
[INFO] sqlitebiter file: convert 'sample_data_multi.json' to 'table_a'↳
↳table
[INFO] sqlitebiter file: convert 'sample_data_multi.json' to 'table_b'↳
↳table
[INFO] sqlitebiter file: convert 'sample_data_single.json' to 'sample_data_
↳single' table
[INFO] sqlitebiter file: convert 'sample_data.xlsx' to 'samplesheet1' table
[INFO] sqlitebiter file: convert 'sample_data.xlsx' to 'samplesheet3' table
[INFO] sqlitebiter file: converted results: source=4, success=6, created-
↳table=6
[INFO] sqlitebiter file: database path: sample.sqlite

```

Output

```
$ sqlite3 sample.sqlite .schema
CREATE TABLE IF NOT EXISTS '_source_info_' ("source_id" INTEGER NOT NULL,
↳ "dir_name" TEXT, "base_name" TEXT NOT NULL, "format_name" TEXT NOT NULL,
↳ "dst_table" TEXT NOT NULL, size INTEGER, mtime INTEGER);
CREATE TABLE IF NOT EXISTS 'sample_data' ("attr_a" INTEGER, "attr_b" REAL,
↳ "attr_c" TEXT);
CREATE TABLE IF NOT EXISTS 'table_a' ("attr_a" INTEGER, "attr_b" REAL,
↳ "attr_c" TEXT);
CREATE TABLE IF NOT EXISTS 'table_b' (a INTEGER, b REAL);
CREATE TABLE IF NOT EXISTS 'sample_data_single' ("attr_a" INTEGER, "attr_b
↳ " REAL, "attr_c" TEXT);
CREATE TABLE IF NOT EXISTS 'samplesheet1' (a INTEGER, b REAL, c TEXT);
CREATE TABLE IF NOT EXISTS 'samplesheet3' (aa INTEGER, ab TEXT, ac TEXT);
```

Designate multiple file path to convert:

Example

```
$ sqlitebiter file sample_data.csv sample_data.xlsx
[INFO] sqlitebiter file: convert 'sample_data.csv' to 'sample_data' table
[INFO] sqlitebiter file: convert 'sample_data.xlsx' to 'samplesheet1' table
[INFO] sqlitebiter file: convert 'sample_data.xlsx' to 'samplesheet3' table
[INFO] sqlitebiter file: converted results: source=2, success=3, created-
↳ table=3
[INFO] sqlitebiter file: database path: out.sqlite
```

Note:

- Available JSON Schema is limited. Acceptable format described in [here](#)
- Wildcard characters cannot use in Windows environments

You could specify converting file format with the `--format` option

Example

```
$ sqlitebiter file --format csv sample_data
[INFO] sqlitebiter file: convert 'sample_data.csv' to 'sample_data' table
[INFO] sqlitebiter file: converted results: source=1, success=1, created-
↳ table=1
[INFO] sqlitebiter file: database path: out.sqlite
```

4.2.2 sqlitebiter file subcommand help

Usage: `sqlitebiter file [OPTIONS] [FILES]...`

Convert tabular data within CSV/Excel/HTML/JSON/Jupyter Notebook/LDJSON/LTSV/Markdown/Mediawiki/SQLite/SSV/TSV file(s) **or** named pipes to a SQLite database file.

Options:

`-r, --recursive` Read **all** files under each directory,

(continues on next page)

(continued from previous page)

```

recursively.
--pattern PATTERN          Convert files matching PATTERN.
--exclude PATTERN         Exclude files matching PATTERN.
--follow-symlinks         Follow symlinks.
-f, --format [csv|excel|html|json|json_
↪lines|jsonl|ldjson|ltsv|markdown|mediawiki|ndjson|sqlite|ssv|tsv|ipynb]
                          Data format to loading (auto-detect from
                          file extensions in default).
--encoding ENCODING       Encoding to load files. Auto-detection from
                          files in default.
-h, --help                Show this message and exit.

Documentation: https://sqlitebiter.rtf.d.io/ Issue tracker:
https://github.com/thombashi/sqlitebiter/issues

```

4.2.3 Supported data formats

Following table shows that the supported data formats:

Table 1: Available data formats

| Format | File Extension | Remarks |
|---------------------|------------------------|--|
| CSV | .csv | |
| Excel | .xlsx/.xls | Create table for each sheet in the Excel workbook. |
| HTML | .html/.htm | Scrape tabular data from <table> tags in the HTML file. And create table for each <table> tag data. |
| JSON | .json | |
| Jupyter Notebook | .ipynb | |
| Line-delimited JSON | .jsonl/.ldjson/.ndjson | |
| LTSV | .ltsv | |
| Markdown | .md | Extract tabular data in the Markdown file. And create a table for each <table> tabular data. |
| SQLite | .sqlite/.sqlite3 | |
| TSV | .tsv | |

4.2.4 Table naming convention

Table name automatically decided as follows for each format:

| Format | Table Name |
|----------|---|
| CSV | <filename> |
| Excel | <Sheet name> |
| HTML | <title>_<key>. <title> replaced with the title tag of the page. <key> replaced with: (1) id attribute of the table tag. (2) unique string if id attribute not present in the table tag. |
| JSON | |
| LTSV | <filename> |
| Markdown | <filename> |
| TSV | <filename> |

- **Common behavior**

- <filename> replaced with filename of converting file (without extensions)

* e.g. If the input file name is sample.csv, <filename> is sample

If a created table name already exists in the database, the behavior differs depending on the existing table (after this referred to as A) and create table (after this referred to as B) structure:

1. **A and B has the same table name and table structure**

- Append creating table data to the existing table data

2. **A and B has the same table name, but different table structure**

- sqlitebiter try to create unique table name for B by appending suffix id number

4.3 Create a SQLite database from URL

sqlitebiter url is a subcommand to fetch table data from the Internet and convert to a SQLite database file.

4.3.1 Example

Following is an example that converts HTML table tags within a web page to SQLite tables by the web page URL.

Example

```
$ sqlitebiter url "https://en.wikipedia.org/wiki/Comparison_of_firewalls"
[INFO] sqlitebiter url: convert 'Comparison_of_firewalls' to 'Comparison_
of_firewalls_Wikipedia_html1' table
[INFO] sqlitebiter url: convert 'Comparison_of_firewalls' to 'Comparison_
of_firewalls_Wikipedia_html2' table
[INFO] sqlitebiter url: convert 'Comparison_of_firewalls' to 'Comparison_
of_firewalls_Wikipedia_html3' table
```

(continues on next page)

(continued from previous page)

```
[INFO] sqlitebiter url: convert 'Comparison_of_firewalls' to 'Comparison_
↳of_firewalls_Wikipedia_html4' table
[INFO] sqlitebiter url: convert 'Comparison_of_firewalls' to 'Comparison_
↳of_firewalls_Wikipedia_html5' table
[INFO] sqlitebiter url: convert 'Comparison_of_firewalls' to 'Comparison_
↳of_firewalls_Wikipedia_html6' table
[INFO] sqlitebiter url: convert 'Comparison_of_firewalls' to 'Comparison_
↳of_firewalls_Wikipedia_html7' table
[INFO] sqlitebiter url: convert 'Comparison_of_firewalls' to 'Comparison_
↳of_firewalls_Wikipedia_html8' table
[INFO] sqlitebiter url: convert 'Comparison_of_firewalls' to 'Comparison_
↳of_firewalls_Wikipedia_html9' table
[INFO] sqlitebiter url: convert 'Comparison_of_firewalls' to 'Comparison_
↳of_firewalls_Wikipedia_html10' table
[INFO] sqlitebiter url: convert 'Comparison_of_firewalls' to 'Comparison_
↳of_firewalls_Wikipedia_html11' table
[INFO] sqlitebiter url: converted results: source=1, success=11, created-
↳table=11
[INFO] sqlitebiter url: database path: out.sqlite
```

Output

```
$ sqlite3 out.sqlite .schema
CREATE TABLE IF NOT EXISTS '_source_info_' ("source_id" INTEGER NOT NULL,
↳"dir_name" TEXT, "base_name" TEXT NOT NULL, "format_name" TEXT NOT NULL,
↳"dst_table" TEXT NOT NULL, size INTEGER, mtime INTEGER);
CREATE TABLE IF NOT EXISTS 'Comparison_of_firewalls_Wikipedia_html1'
↳(Firewall TEXT, License TEXT, [Cost and usage limits] TEXT, OS TEXT);
CREATE TABLE IF NOT EXISTS 'Comparison_of_firewalls_Wikipedia_html2'
↳(Firewall TEXT, License TEXT, Cost TEXT, OS TEXT);
CREATE TABLE IF NOT EXISTS 'Comparison_of_firewalls_Wikipedia_html3' ([Can_
↳Target:] TEXT, [Changing default policy to accept/reject (by issuing a
↳single rule)] TEXT, [IP destination address(es)] TEXT, [IP source_
↳address(es)] TEXT, [TCP/UDP destination port(s)] TEXT, [TCP/UDP source_
↳port(s)] TEXT, [Ethernet MAC destination address] TEXT, [Ethernet MAC_
↳source address] TEXT, [Inbound firewall (ingress)] TEXT, [Outbound_
↳firewall (egress)] TEXT);
CREATE TABLE IF NOT EXISTS 'Comparison_of_firewalls_Wikipedia_html4'
↳([Can:] TEXT, [work at OSI Layer 4 (stateful firewall)] TEXT, [work at_
↳OSI Layer 7 (application inspection)] TEXT, [Change TTL? (Transparent to_
↳traceroute)] TEXT, [Configure REJECT-with answer] TEXT, [DMZ (de-
↳militarized zone) - allows for single/several hosts not to be firewalled.
↳] TEXT, [Filter according to time of day] TEXT, [Redirect TCP/UDP ports_
↳(port forwarding)] TEXT, [Redirect IP addresses (forwarding)] TEXT,
↳[Filter according to User Authorization] TEXT, [Traffic rate-limit /_
↳QoS] TEXT, Tarpit TEXT, Log TEXT);
CREATE TABLE IF NOT EXISTS 'Comparison_of_firewalls_Wikipedia_html5'
↳([Features:] TEXT, "Configuration: GUI_ text or both modes?" TEXT,
↳"Remote Access: Web (HTTP)_ Telnet_ SSH_ RDP_ Serial COM RS232_ ..."
↳TEXT, [Change rules without requiring restart?] TEXT, [Ability to_
↳centrally manage all firewalls together] TEXT);
CREATE TABLE IF NOT EXISTS 'Comparison_of_firewalls_Wikipedia_html6'
↳([Features:] TEXT, [Modularity: supports third-party modules to extend
↳functionality?] TEXT, [IPS : Intrusion prevention system] TEXT, [Open-
↳Source License?] TEXT, [supports IPv6 ?] TEXT, [Class: Home /_
↳Professional] TEXT, [Operating Systems on which it runs?] TEXT);
```

(continues on next page)

(continued from previous page)

```

CREATE TABLE IF NOT EXISTS 'Comparison_of_firewalls_Wikipedia_html7' (
↳([Can:] TEXT, "NAT44 (static_ dynamic w/o ports_ PAT)" TEXT, "NAT64_
↳NPTv6" TEXT, [IDS (Intrusion Detection System)] TEXT, [VPN (Virtual
↳Private Network)] TEXT, [AV (Anti-Virus)] TEXT, Sniffer TEXT, [Profile
↳selection] TEXT);
CREATE TABLE IF NOT EXISTS 'Comparison_of_firewalls_Wikipedia_html8' (
↳([vteFirewall software] TEXT);
CREATE TABLE IF NOT EXISTS 'Comparison_of_firewalls_Wikipedia_html9' (A
↳TEXT, B TEXT);
CREATE TABLE IF NOT EXISTS 'Comparison_of_firewalls_Wikipedia_html10' (A
↳TEXT, B TEXT);
CREATE TABLE IF NOT EXISTS 'Comparison_of_firewalls_Wikipedia_html11' (A
↳TEXT, B TEXT);

```

The attributes within the converted SQLite database may include symbols as the above. Symbols within attributes can be replaced by using `--replace-symbol` option. In the following example shows replace symbols to underscores.

Example

```

$ sqlitebiter --replace-symbol _ -q url "https://en.wikipedia.org/wiki/
↳Comparison_of_firewalls"

```

Output

```

$ sqlite3 out.sqlite .schema
CREATE TABLE IF NOT EXISTS '_source_info_' ("source_id" INTEGER NOT NULL,
↳"dir_name" TEXT, "base_name" TEXT NOT NULL, "format_name" TEXT NOT NULL,
↳"dst_table" TEXT NOT NULL, size INTEGER, mtime INTEGER);
CREATE TABLE IF NOT EXISTS 'Comparison_of_firewalls_Wikipedia_html1' (
↳(Firewall TEXT, License TEXT, "Cost_and_usage_limits" TEXT, OS TEXT);
CREATE TABLE IF NOT EXISTS 'Comparison_of_firewalls_Wikipedia_html2' (
↳(Firewall TEXT, License TEXT, Cost TEXT, OS TEXT);
CREATE TABLE IF NOT EXISTS 'Comparison_of_firewalls_Wikipedia_html3' ("Can
↳Target" TEXT, "Changing_default_policy_to_accept_reject_by_issuing_a
↳single_rule" TEXT, "IP_destination_address_es" TEXT, "IP_source_address_
↳es" TEXT, "TCP_UDP_destination_port_s" TEXT, "TCP_UDP_source_port_s"
↳TEXT, "Ethernet_MAC_destination_address" TEXT, "Ethernet_MAC_source_
↳address" TEXT, "Inbound_firewall_ingress" TEXT, "Outbound_firewall_egress
↳" TEXT);
CREATE TABLE IF NOT EXISTS 'Comparison_of_firewalls_Wikipedia_html4' (Can
↳TEXT, "work_at_OSI_Layer_4_stateful_firewall" TEXT, "work_at_OSI_Layer_7_
↳application_inspection" TEXT, "Change_TTL_Transparent_to_traceroute"
↳TEXT, "Configure_REJECT_with_answer" TEXT, "DMZ_de_militarized_zone_
↳allows_for_single_several_hosts_not_to_be_firewalled" TEXT, "Filter_
↳according_to_time_of_day" TEXT, "Redirect_TCP_UDP_ports_port_forwarding"
↳TEXT, "Redirect_IP_addresses_forwarding" TEXT, "Filter_according_to_User_
↳Authorization" TEXT, "Traffic_rate_limit_QoS" TEXT, Tarpit TEXT, Log
↳TEXT);
CREATE TABLE IF NOT EXISTS 'Comparison_of_firewalls_Wikipedia_html5' (
↳(Features TEXT, "Configuration_GUI_text_or_both_modes" TEXT, "Remote_
↳Access_Web_HTTP_Telnet_SSH_RDP_Serial_COM_RS232" TEXT, "Change_rules_
↳without_requiring_restart" TEXT, "Ability_to_centrally_manage_all_
↳firewalls_together" TEXT);

```

(continues on next page)

(continued from previous page)

```

CREATE TABLE IF NOT EXISTS 'Comparison_of_firewalls_Wikipedia_html6' (
↳(Features TEXT, "Modularity_supports_third_party_modules_to_extend_
↳functionality" TEXT, "IPS _Intrusion_prevention_system" TEXT, "Open_
↳Source_License" TEXT, "supports_IPv6" TEXT, "Class_Home_Professional"
↳TEXT, "Operating_Systems_on_which_it_runs" TEXT);
CREATE TABLE IF NOT EXISTS 'Comparison_of_firewalls_Wikipedia_html7' (Can_
↳TEXT, "NAT44_static_dynamic_w_o_ports_PAT" TEXT, "NAT64_NPTv6" TEXT,
↳"IDS_Intrusion_Detection_System" TEXT, "VPN_Virtual_Private_Network"
↳TEXT, "AV_Anti_Virus" TEXT, Sniffer TEXT, "Profile_selection" TEXT);
CREATE TABLE IF NOT EXISTS 'Comparison_of_firewalls_Wikipedia_html8' (
↳"vteFirewall_software" TEXT);
CREATE TABLE IF NOT EXISTS 'Comparison_of_firewalls_Wikipedia_html9' (A_
↳TEXT, B TEXT);
CREATE TABLE IF NOT EXISTS 'Comparison_of_firewalls_Wikipedia_html10' (A_
↳TEXT, B TEXT);
CREATE TABLE IF NOT EXISTS 'Comparison_of_firewalls_Wikipedia_html11' (A_
↳TEXT, B TEXT);

```

4.3.2 sqlitebiter url subcommand help

Usage: sqlitebiter url [OPTIONS] URL

Scrape tabular data **from a URL and** convert data to a SQLite database file.

Options:

```

-f, --format [csv|excel|html|json|json_
↳lines|jsonl|ldjson|ltsv|markdown|mediawiki|ndjson|sqlite|ssv|tsv|ipy nb]
      Data format to loading (defaults to html).
-e, --encoding ENCODING      HTML page read encoding. Defaults to utf-8.
-p, --proxy PROXY            Specify a proxy in the form
                              [user:passwd@]proxy.server:port.
-h, --help                    Show this message and exit.

```

Documentation: <https://sqlitebiter.rtfid.io/> Issue tracker:
<https://github.com/thombashi/sqlitebiter/issues>

4.4 Create a SQLite database from stdin

`sqlitebiter stdin` is a subcommand to convert tabular data text from stdin to a SQLite database file.

4.4.1 Examples

A data format is a mandatory argument for `sqlitebiter stdin subcommand`:

Example

```
$ echo '[{"hoge": 4, "foo": "abc"}, {"hoge": 12, "foo": "xyz"}]' |
↳ sqlitebiter stdin json
[INFO] convert 'stdin' to 'json1' table
[INFO] converted results: source=1, success=1, created-table=1
[INFO] database path: out.sqlite
```

4.4.2 sqlitebiter stdin subcommand help

```
Usage: sqlitebiter stdin [OPTIONS] {csv|html|json|json_lines|jsonl|ldjson|ltsv
|markdown|mediawiki|ndjson|ssv|tsv|ipynb}
```

Convert tabular data within CSV/HTML/JSON/Jupyter Notebook/LDJSON/LTSV/Markdown/Mediawiki/SSV/TSV text to a SQLite database file.

Options:

`-h, --help` Show this message **and** exit.

Documentation: <https://sqlitebiter.rtf.d.io/> Issue tracker: <https://github.com/thombashi/sqlitebiter/issues>

4.5 Create a SQLite database from Google Sheets

`sqlitebiter gs` is a subcommand to convert [Google Sheets](#) to a SQLite database file.

4.5.1 Requirements

Following python packages are required to use Google Sheets feature.

- `gsread`
- `oauth2client`
- `pyOpenSSL`

Dependency Python package installation:

```
$ pip install sqlitebiter[gs]
```

Or

```
$ sudo pip install gspread
$ sudo pip install oauth2client pyopenssl
```

4.5.2 Example

Example

```
$ sqlitebiter -o sample.sqlite gs credentials-xxxxxxxxxxxx.json samplebook
[INFO] sqlitebiter gs: convert 'samplebook' to 'sheet3' table
[INFO] sqlitebiter gs: convert 'samplebook' to 'sheet1' table
[INFO] sqlitebiter gs: converted results: source=1, success=2, created-
↳ table=2
[INFO] sqlitebiter gs: database path: sample.sqlite
```

Output

```
$ sqlite3 sample.sqlite .schema
CREATE TABLE sqlite_sequence(name,seq);
CREATE TABLE IF NOT EXISTS '_source_info_' ("source_id" INTEGER NOT NULL,
↳ "dir_name" TEXT, "base_name" TEXT NOT NULL, "format_name" TEXT NOT NULL,
↳ "dst_table" TEXT NOT NULL, size INTEGER, mtime INTEGER);
CREATE TABLE IF NOT EXISTS 'sheet3' (a INTEGER, b REAL, c TEXT);
CREATE TABLE IF NOT EXISTS 'sheet1' (a INTEGER, b REAL, c TEXT);
```

4.5.3 sqlitebiter gs subcommand help

```
Usage: sqlitebiter gs [OPTIONS] CREDENTIALS TITLE
```

Convert a spreadsheet **in** Google Sheets to a SQLite database file.

CREDENTIALS: OAuth2 Google credentials file. TITLE: Title of the Google Sheets to convert.

Options:

-h, --help Show this message **and** exit.

Documentation: <https://sqlitebiter.rtfid.io/> Issue tracker:
<https://github.com/thombashi/sqlitebiter/issues>

CHANGELOG

<https://github.com/thombashi/sqlitebiter/releases>

SPONSORS



Become a sponsor

INDICES AND TABLES

- `genindex`

**CHAPTER
EIGHT**

LINKS

- [pip](#): A tool for installing python packages
- [GitHub repository](#)
- [Issue tracker](#)

INDICES AND TABLES

- `genindex`