toml-sort

Release 0.17.0

Samuel Roeca

Aug 31, 2022

TABLE OF CONTENTS

See below for a the command line interface.

Toml sort: a sorting utility for toml files.

```
usage: toml-sort [-h] [--version] [-o OUTPUT] [-a] [-i] [--no-header]
                 [--check] [-I]
                 [F [F ...]]
```

CHAPTER

ONE

POSITIONAL ARGUMENTS

F filename(s) to be processed by toml-sort (default: -)

CHAPTER

TWO

NAMED ARGUMENTS

--version	display version information and exit
	Default: False
-o, --output	output filepath (default: '-')
-a, --all	sort ALL keys (default: only sort non-inline 'tables and arrays of tables')
	Default: False
-i, --in-place	overwrite the original input file with changes
	Default: False
--no-header	do not keep a document's leading comments
	Default: False
--check	silently check if a file's contents would be changed by the formatter
	Default: False
-I, --ignore-case	ignore case when sorting
	Default: False

Examples:

- **Stdin -> Stdout**: cat input.toml | toml-sort
- **Disk -> Disk**: toml-sort -o output.toml input.toml
- **Linting**: toml-sort –check input.toml input2.toml input3.toml
- **Inplace Disk**: toml-sort –in-place input.toml input2.toml

Return codes:

- 0 : success.
- 1 : errors were found

Notes:

- You cannot redirect from a file to itself in Bash. POSIX shells process redirections first, then execute commands. –in-place exists for this reason

CHAPTER

THREE

MODULES

This page documents importable modules and their contents.

3.1 toml_sort.tomlsort

Utility functions and classes to sort toml text.

class toml_sort.tomlsort.**TomlSort**(*input_toml: str*, *only_sort_tables: bool = False*, *no_header: bool = False*, *ignore_case: bool = False*)

API to manage sorting toml files.

Variables

- **input_toml** (*str*) – the input toml data for processing
- **only_sort_tables** (*bool*) – turns on sorting for only tables
- **no_header** (*bool*) – omit leading comments from the output
- **sort_func** (*callable*) – the sorting function to use for lists of items

sorted() → str

Sort a TOML string.

sorted_children_table(*parent: Union[tomlkit.items.Table, tomlkit.toml_document.TOMLDocument]*) → Iterable[Tuple[str, tomlkit.items.Item]]

Get the sorted children of a table.

NOTE: non-tables are wrapped in an item to ensure that they are, in fact, items. Tables and AoT's are definitely items, so the conversion is not necessary.

toml_doc_sorted(*original: tomlkit.toml_document.TOMLDocument*) → tomlkit.toml_document.TOMLDocument

Sort a TOMLDocument.

toml_elements_sorted(*original: tomlkit.items.Item*) → tomlkit.items.Item

Returns a sorted item, recursing collections to their base.

CHAPTER

FOUR

TOML-SORT

A command line utility to sort and format your toml files.

Read the latest documentation here: https://toml-sort.readthedocs.io/en/latest/

4.1 Installation

```
# With pip
pip install toml-sort

# With poetry
poetry add --dev toml-sort
```

4.2 Motivation

This library sorts TOML files, providing the following features:

- Sort tables and Arrays of Tables (AoT)
- Option to sort non-tables / non-AoT's, or not
- Preserve inline comments
- Option to preserve top-level document comments, or not
- Standardize whitespace and indentation

I wrote this library/application because I couldn't find any "good" sorting utilities for TOML files. Now, I use this as part of my daily workflow. Hopefully it helps you too!

4.3 Command line usage

This project can be used as either a command line utility or a Python library. Read the docs for an overview of its library capabilities. For command line usage, see below:

```
$ toml-sort --help
Usage: toml-sort [OPTIONS] [FILENAMES]...

  Sort toml file FILENAME(s), writing to file(s) or stdout (default)

  FILENAME a filepath or standard input (-)

  Examples (non-exhaustive list):
    Stdin -> Stdout : cat input.toml | toml-sort
    Disk -> Disk    : toml-sort -o output.toml input.toml
    Linting         : toml-sort --check input.toml input2.toml input3.toml
    Inplace Disk    : toml-sort --in-place input.toml input2.toml

Options:
  -o, --output PATH  The output filepath. Choose stdout with '-' (the
                     default).

  -a, --all          Sort all keys. Default is to only sort non-inline 'tables
                     and arrays of tables'.

  -i, --in-place     Makes changes to the original input file. Note: you
                     cannot redirect from a file to itself in Bash. POSIX
                     shells process redirections first, then execute the
                     command.

  --no-header        Do not keep a document's leading comments.
  --check            Check if an original file is changed by the formatter.
                     Return code 0 means it would not change. Return code 1
                     means it would change.

  -I, --ignore-case  When sorting, ignore case.
  --version          Show the version and exit.
  --help             Show this message and exit.
```

4.4 Configuration file

toml-sort can also be configured by using the `pyproject.toml` file. If the file exists and has a `tool.tomlsort` section, the configuration is used. If both command line arguments and the configuration are used, the options are merged. In the case of conflicts, the command line option is used.

In short, the names are the same as on the command line (and have the same meaning), but `-` is replaced with `_`. Please note, that only the below options are supported:

```
[tool.tomlsort]
all = true
in_place = true
```

(continues on next page)

(continued from previous page)

```
no_header = true
check = true
ignore_case = true
```

4.5 Example

The following example shows the input, and output, from the CLI with default options.

4.5.1 Unformatted, unsorted input

```
# My great TOML example

  title = "The example"

[[a-section.hello]]
ports = [ 8001, 8001, 8002 ]
dob = 1979-05-27T07:32:00Z # First class dates? Why not?



  [b-section]
  date = "2018"
  name = "Richard Stallman"

[[a-section.hello]]
ports = [ 80 ]
dob = 1920-05-27T07:32:00Z # Another date!

                           [a-section]
                           date = "2019"
                           name = "Samuel Roeca"
```

4.5.2 Formatted, sorted output

```
# My great TOML example

title = "The example"

[a-section]
date = "2019"
name = "Samuel Roeca"

[[a-section.hello]]
ports = [ 8001, 8001, 8002 ]
dob = 1979-05-27T07:32:00Z # First class dates? Why not?

[[a-section.hello]]
```

(continues on next page)

(continued from previous page)

```
ports = [ 80 ]
dob = 1920-05-27T07:32:00Z # Another date!

[b-section]
date = "2018"
name = "Richard Stallman"
```

4.6 Local Development

Local development for this project is quite simple.

Dependencies

Install the following tools manually.

- Poetry>=1.0
- GNU Make

Recommended

- asdf

Set up development environment

```
make setup
```

Run Tests

```
make test
```

4.7 Written by

Samuel Roeca, *samuel.roeca@gmail.com*

CHAPTER

FIVE

INDICES AND TABLES

- genindex
- modindex

PYTHON MODULE INDEX

t

INDEX

M

S

T