Frostmark Documentation

Peter Badida

May 05, 2019

Contents:

1 Getting started **1**
- 1.1 Requirements . . . 1
- 1.2 Installation . . . 1
- 1.3 Commands . . . 2

2 Config **3**
- 2.1 React GUI . . . 3

3 Entrypoint **5**

4 Common **7**

5 DB Base **9**

6 Database **11**

7 Models **13**

8 User data **15**

9 Parser **17**

10 Profiles **19**

11 Importer **21**

12 Exporter **23**

13 Core **25**

14 Tests **27**
- 14.1 Test: Database . . . 27

15 Indices and tables **29**

Python Module Index **31**

CHAPTER 1

Getting started

1.1 Requirements

- Python 3.7+ (however 3.6 should work too)

1.2 Installation

Frostmark is distributed as a Python package on PyPI and on GitHub as a source release with packages included. You can install multiple versions:

- CLI
- React GUI

The `CLI` version is the basic building stone and should work just fine for quick listing or pulling and exporting the bookmarks from the browsers into some common format. While you can use advanced options for editing bookmarks, folders and their relationships, using the `React GUI` option is prettier.

You can install the `CLI` with:

```
pip install frostmark
```

For the `React GUI` version you need to specify an additional flag for `pip` to install even the GUI dependencies:

```
pip install frostmark[gui_react]
```

For installing from files (`.tar.gz` or `.whl`) you need to adjust the name a little bit:

```
# archive
pip install frostmark.tar.gz
pip install frostmark.tar.gz[gui_react]

# wheel package
```

(continues on next page)

(continued from previous page)

```
pip install frostmark.whl
pip install frostmark.whl[gui_react]
```

1.3 Commands

After the successful installation you are provided with multiple options for accessing the program:

- From Python as a module:

```
# module help
python -m frostmark
python -m frostmark --help

# CLI & CLI help
python -m frostmark console
python -m frostmark console --help

# GUI & GUI help
python -m frostmark gui
python -m frostmark gui --help
```

- By specific commands provided by frostmark to the OS:

```
# CLI & CLI help
fmcli
fmcli --help

# GUI & GUI help
fmgui
fmgui --help
```

CHAPTER 2

Config

Modify how the specific parts of the application run with the Environment variables. The values have to be set before the application is started.

2.1 React GUI

FROSTMARK_HOST Specify what host you want to run the server layer between Frostmark and React frontend displaying the bookmarks.

```
FROSTMARK_HOST=192.168.1.100 fmgui
```

2.1.1 Development

REACT_PROXY Specify the endpoint for all the requests attempted to fetch from path starting with `/api` via Node's `fetch()`. For more info see the proxy config for specific routes in `src/setupProxy.js`

```
REACT_PROXY=http://192.168.1.100:5000 ./run_frontend.sh
```

Note: Check firewall rules in case of not being able to proxy the requests.

CHAPTER 3

Entrypoint

Module for interfacing with the package via python command:

```
python -m frostmark
```

CHAPTER 4

Common

Module for common functions.

CHAPTER 5

DB Base

Module for creating a connection to SQLite DB and SQLAlchemy declarative base home.

CHAPTER 6

Database

Module for creating SQLite DB schema and session retrieving.

frostmark.db.**folder_check_root**(*session*)
Check whether default rows for Folder model exist and create them if not.

frostmark.db.**get_session**()
Create DB schema and return a new session.

CHAPTER 7

Models

ORM SQLAlchemy models for Frostmark application storage.

class frostmark.models.**Bookmark**(***kwargs*)
Bookmark item containing URL, title, icon (favicon.ico) and folder_id.

class frostmark.models.**Folder**(***kwargs*)
Folder item for grouping bookmarks. ID 0 is reserved for ROOT folder.

get_root
Return root folder constants to e.g. create root folder value in DB if it does not exist.

CHAPTER 8

User data

CHAPTER 9

Parser

Module for parsing arguments from console.

class frostmark.parser.**ExecuteAction**(**a*, *func=None*, *func_args=()*, *func_kwargs={}*, ***kw*)

ArgumentParser action for add_argument(action=...)

Executes a function and its args + kwargs passed as the __init__ arguments:

- func
- func_args
- func_kwargs

together with argument values inserted into func_kwargs as 'arg_values' and exits immediately.

class frostmark.parser.**FrostmarkArgumentParser**(*prog=None*, *usage=None*, *description=None*, *epilog=None*, *parents=[]*, *formatter_class=<class 'argparse.HelpFormatter'>*, *prefix_chars='-'*, *fromfile_prefix_chars=None*, *argument_default=None*, *conflict_handler='error'*, *add_help=True*, *allow_abbrev=True*)

Inheriting from *ArgumentParser* to print custom print message.

CHAPTER 10

Profiles

Module for retrieving all ‘profiles’ from specified browser.

CHAPTER 11

Importer

Import bookmarks from various bookmarks database files into internal database.

class frostmark.importer.**Importer**

Interface for retrieving and importing bookmarks into local storage for multiple backends.

import_from

Import bookmarks from particular path into internal storage.

CHAPTER 12

Exporter

Export bookmarks from internal database into various formats.

class frostmark.exporter.**Exporter**

Interface for retrieving and exporting bookmarks from local storage to multiple backends.

export_to

Public export function for STDOUT or file.

prepare_export

Export bookmarks from internal storage.

CHAPTER 13

Core

CHAPTER 14

Tests

14.1 Test: Database

Module for creating a connection to SQLite DB and SQLAlchemy declarative base home.

class frostmark.tests.test_database.**DBBaseTestCase**(*methodName='runTest'*)

TestCase for SQLite database backend.

test_db_base()

Test all variables, locations, files for creating DB base.

test_db_create()

Test creating SQLite3 DB only (without tables) in the user_data section.

test_db_session()

Test getting session for SQLite DB which passively creates a new DB (new .sqlite file) if it isn't present yet.

CHAPTER 15

Indices and tables

- genindex
- modindex
- search

Python Module Index

f

frostmark.__main__, 5
frostmark.common, 7
frostmark.core, 25
frostmark.db, 11
frostmark.db_base, 9
frostmark.exporter, 23
frostmark.importer, 21
frostmark.models, 13
frostmark.parser, 17
frostmark.profiles, 19
frostmark.tests.test_database, 27
frostmark.user_data, 15

Index

B

Bookmark (*class in frostmark.models*), 13

D

DBBaseTestCase (*class in frostmark.tests.test_database*), 27

E

ExecuteAction (*class in frostmark.parser*), 17
export_to (*frostmark.exporter.Exporter attribute*), 23
Exporter (*class in frostmark.exporter*), 23

F

Folder (*class in frostmark.models*), 13
folder_check_root() (*in module frostmark.db*), 11
frostmark.__main__ (*module*), 5
frostmark.common (*module*), 7
frostmark.core (*module*), 25
frostmark.db (*module*), 11
frostmark.db_base (*module*), 9
frostmark.exporter (*module*), 23
frostmark.importer (*module*), 21
frostmark.models (*module*), 13
frostmark.parser (*module*), 17
frostmark.profiles (*module*), 19
frostmark.tests.test_database (*module*), 27
frostmark.user_data (*module*), 15
FrostmarkArgumentParser (*class in frostmark.parser*), 17

G

get_root (*frostmark.models.Folder attribute*), 13
get_session() (*in module frostmark.db*), 11

I

import_from (*frostmark.importer.Importer attribute*), 21
Importer (*class in frostmark.importer*), 21

P

prepare_export (*frostmark.exporter.Exporter attribute*), 23

T

test_db_base() (*frostmark.tests.test_database.DBBaseTestCase method*), 27
test_db_create() (*frostmark.tests.test_database.DBBaseTestCase method*), 27
test_db_session() (*frostmark.tests.test_database.DBBaseTestCase method*), 27