

---

# **treeopt Documentation**

***Release 0.0.1***

**joergbrech**

**Apr 23, 2021**



**CONTENTS:**

- 1 Getting Started 3**
  - 1.1 Installing the Library . . . . . 3
- 2 API Documentation 5**
- 3 Development 7**
  - 3.1 Getting Started . . . . . 7
  - 3.2 Contribution Guide . . . . . 8
  - 3.3 Using the *Makefile* . . . . . 9
  - 3.4 Publishing the Package . . . . . 11
  - 3.5 Building the Documentation . . . . . 11
- 4 Indices and tables 13**
- 5 Dependencies 15**
  - 5.1 requirements.txt . . . . . 15
  - 5.2 Runtime Dependencies and Licenses . . . . . 16
- 6 Indices and tables 17**
- Python Module Index 19**
- Index 21**



A python module for simulation-based optimization



## GETTING STARTED

### 1.1 Installing the Library

You can use `pip` to install *treeopt*.

```
pip install treeopt
```





## API DOCUMENTATION

A python module for simulation-based optimization.



## DEVELOPMENT

### 3.1 Getting Started

This section provides instructions for setting up your development environment. If you follow the steps from top to bottom you should be ready to roll by the end.

#### 3.1.1 Get the Source

The source code for the *TREEOPT* project lives at [github](https://github.com/hbrs-cse/TREEOPT). You can use *git clone* to get it.

```
git clone https://github.com/hbrs-cse/TREEOPT
```

#### 3.1.2 Create the Virtual Environment

You can create a virtual environment and install the project's dependencies using *make*.

```
make venv  
make install  
source venv/bin/activate
```

#### 3.1.3 Try It Out

One way to test out the environment is to run the tests. You can do this with the *make test* target.

```
make test
```

If the tests run and pass, you're ready to roll.

#### 3.1.4 Getting Answers

Once the environment is set up, you can perform a quick build of this project documentation using the *make answers* target.

```
make answers
```

## 3.2 Contribution Guide

Begin by reading the section *Getting Started*. Make sure you are on your master branch and your local copy is up-to-date:

```
git checkout master
git pull
```

### 3.2.1 Create a new feature branch

If you would like to contribute, begin by creating a feature branch

```
git checkout -b jk/issue-42-my-awesome-new-feature
```

Ideally, the name of the branch should have a speaking name. It is good practice to prepend a private branch with your initials. If you are writing a feature that is referenced in an issue on Github, include the issue number in your branch name.

Now implement your modifications. The source code of the library is contained in the subdirectory `treeopt`. Any new feature should be thoroughly tested, so please add unit tests in the `tests` subdirectory.

### 3.2.2 Autoformat your code

Your code style should conform to the PEP. We use `flake8` to check the code at every commit and decline contributions that are not conform to the standard. Using a combination of the tools `pre-commit`, `black` and `flake8` you can automatically re-format and check your code with every commit. To activate this optional feature, install `pre-commit`

```
pip install pre-commit
```

and install the pre-configured commit-hook for this repository. In the root directory of this repo, type

```
pre-commit install
```

That's all there is to it!

### 3.2.3 Commit your changes

```
git commit -m 'implemented an awesome new feature
fixes #42'
```

The commit message should be a short, but good description of the changes. Make as many commits as you need and try to change only one thing. A good rule of thumb is, that using the word “and” in a commit message is a sign, that you could have used two commits instead.

You can add a reference to an Github issue number using a hashtag.

### 3.2.4 Push your local branch

```
git push -u origin HEAD
```

This will generate a new branch in the remote repository with the same name as your local branch and the sync the two branches. After the remote branch has been created, the command

```
git push
```

suffices to push your local changes to the remote branch.

### 3.2.5 Create a Pull Request

If you are satisfied with your changes, navigate to the repo on Github and open a Pull Request. You can prepend the Pull Request name with *[WIP]* to label it as “Work in Progress”. This way, noone will prematurely review your unfinished work.

## 3.3 Using the *Makefile*

This project includes a [Makefile](#) that you can use to perform common tasks such as running tests and building documentation.

### 3.3.1 Targets

This section contains a brief description of the targets defined in the `Makefile`.

#### `clean`

Remove generated packages, documentation, temporary files, *etc.*

#### `lint`

Run [pylint](#) against the project files.

#### `test`

Run the unit tests.

#### `quicktest`

Run the unit tests without performing pre-test validations (like *linting*).

### docs

Build the documentation for production.

---

**Note:** You can also build the documents directly, bypassing validations like *linting* and *testing* using [Sphinx Makefile](#) directly.

```
cd docs
make clean && make html
make latexpdf
```

---

### answers

Perform a quick build of the documentation and open it in your browser.

### package

Build the package for publishing.

### publish

Publish the package to your repository.

### build

Install the current project locally so that you may run the command-line application.

### venv

Create a virtual environment.

### install

Install (or update) project dependencies.

### licenses

Generate a report of the projects dependencies and respective licenses.

---

**Note:** If project dependencies change, please update this documentation.

---

## 3.4 Publishing the Package

As you make changes to the project, you'll probably want to publish new version of the package. (*That's the point, right?*)

### 3.4.1 Publishing

The actual process of publishing the project is just a matter of running the *publish* target.

```
make publish
```

### 3.4.2 Installing

If you just need to install the library in your project, have a look at the *general tutorial* article.

## 3.5 Building the Documentation

### 3.5.1 Sphinx

The documentation in this project is generated by *Sphinx* from *reStructuredTex*.

### 3.5.2 Ubuntu/Debian

This project started on *Ubuntu Linux 18.04*. That doesn't mean you can't use another distribution, or even another operating system, but you may need to perform some additional setup steps to get your builds working. (If you do get it working under another system, please consider adding an article to let others know how you did it!)

#### Prerequisites

The project uses the Sphinx *LatexBuilder* to generate a *PDF* document. If you're using Ubuntu (or Debian) you'll need to install *texlive* and *latexmk*.

```
sudo apt-get install texlive-latex-recommended \
    texlive-latex-extra \
    texlive-fonts-recommended \
    latexmk
```

### 3.5.3 make

Once everything is in place, you can build the documentation using the *make docs* the target defined in the project's *Makefile*.

```
make docs
```





## INDICES AND TABLES

- `genindex`
- `modindex`
- `search`



## DEPENDENCIES

The `requirements.txt` file contains this project's module dependencies. You can install these dependencies using `pip`.

```
pip install -r requirements.txt
```

### 5.1 requirements.txt

```
click>=7.0,<8
pip-check-reqs>=2.0.1,<3
pip-licenses==3.3.1

flake8>=3.7.9,<4
flake8-docstrings>=1.5.0,<2
pytest==6.2.3
pytest-cov>=2.5.1,<3
pytest-pythonpath>=0.7.2,<1
setuptools>=38.4.0
Sphinx>=2.2.0
sphinx-rtd-theme>=0.4.3,<1
tox>=3.0.0,<4
twine==3.4.1
pre-commit>=2.2.0
```

## 5.2 Runtime Dependencies and Licenses

Name	Version	License	URL
Click	7.0	BSD	<a href="https://palletsprojects.com/p/click/">https://palletsprojects.com/p/click/</a>
bleach	3.1.0	Apache Software License	<a href="https://github.com/mozilla/bleach">https://github.com/mozilla/bleach</a>
filelock	3.0.12	Public Domain < <a href="http://unlicense.org">http://unlicense.org</a> >	<a href="https://github.com/benediktschmitt/py-filelock">https://github.com/benediktschmitt/py-filelock</a>
frozensorteddict	1.2.0	MIT	<a href="https://github.com/wsmith323/frozensorteddict">https://github.com/wsmith323/frozensorteddict</a>
importlib-metadata	0.18	Apache Software License	<a href="http://importlib-metadata.readthedocs.io/">http://importlib-metadata.readthedocs.io/</a>
readme-renderer	24.0	Apache License, Version 2.0	<a href="https://github.com/pypa/readme_renderer">https://github.com/pypa/readme_renderer</a>
toml	0.10.0	MIT	<a href="https://github.com/uiri/toml">https://github.com/uiri/toml</a>
webencodings	0.5.1	BSD	<a href="https://github.com/SimonSapin/python-webencodings">https://github.com/SimonSapin/python-webencodings</a>
zipp	0.5.1	UNKNOWN	<a href="https://github.com/jaraco/zipp">https://github.com/jaraco/zipp</a>

## INDICES AND TABLES

- `genindex`
- `modindex`
- `search`



## PYTHON MODULE INDEX

### t

`treeopt`, 5





## INDEX

### M

module  
    treeopt, 5

### T

treeopt  
    module, 5