# gols

*Release 1.0.0*

Sep 07, 2017

# Contents


**1 Overview** **1**
- 1.1 Installation . . . 1
- 1.2 Documentation . . . 1
- 1.3 Development . . . 1

**2 Installation** **3**

**3 Usage** **5**
- 3.1 The CLI way . . . 5
- 3.2 The automatic way . . . 5

**4 Reference** **7**
- 4.1 gols . . . 7

**5 Contributing** **9**
- 5.1 Bug reports . . . 9
- 5.2 Documentation improvements . . . 9
- 5.3 Feature requests and feedback . . . 9
- 5.4 Development . . . 10

**6 Authors** **11**

**7 Changelog** **13**
- 7.1 0.1.0 (2017-09-06) . . . 13

**8 Indices and tables** **15**

**Python Module Index** **17**

CHAPTER 1

# Overview

| | |
|---|---|
| docs | |
| tests | |
| package | |

gols

- Free software: BSD license

## Installation

```
pip install gols
```

## Documentation

https://gols.readthedocs.io/

## Development

To run the all tests run:

```
tox
```

Note, to combine the coverage data from all the tox environments run:

| | |
|---|---|
| Windows | `set PYTEST_ADDOPTS=--cov-append tox` |
| Other | `PYTEST_ADDOPTS=--cov-append tox` |

CHAPTER 2

# Installation

At the command line:

```
pip install gols
```

CHAPTER 3

# Usage

## The CLI way

You can use gols with your favorite command line terminal, that's fine, just run *gols –help* and see what commands, switches you need to enter !

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

Options:
  --debug / --no_debug  Set to true to see debug logs on top of info
  --help                Show this message and exit.

Commands:
  upload  uploads .fit files to your garmin connect account
```

The upload command needs in particular your username and password account. You can pass them with the *-u* and *-p* switches respectively, however it's recommended you use the environment variables. So either you do

```
export GARMINCONNECT_USERNAME=user GARMINCONNECT_PASSWORD=password gols --debug␣
↪upload blabblablabla
```

Either you add those variables in your favorite .zshrc, if you just have a .bashrc go get zsh and oh-my-zsh !

## The automatic way

To use gols in a more user friendly manner, you'll need 3 things, this assumes your distribution uses systemd, should it not be the case you can adapt using udev rules ! The installation is a little-bit involved but worth the effort, I now just plugs my watch and bam, it's uploaded !

1. Add your Garmin device to your /etc/fstab, mine for instance

```
#garmin fenix 2
UUID=489A-9E97 /media/fenix2 vfat auto,nofail,rw,user,uid=1000,gid=1000 0␣
↪2
```

Then endpoint */media/fenix2* is created by the user who will upload its activities and you'll have to run *sudo blkid* to get the device UUID. Issue a *systemd daemon-reload* so that you get the mount name systemd will assign to your new entry.

2. Create a systemd user unit with *systemctl –user edit gols.service –force*.

What is important in that file is the *media-fenix2.mount*, adapt yours with what systemd came up after step 1.

```
[Unit]
Description=gols a little less now
Requires=media-fenix2.mount
After=media-fenix2.mount
[Service]
Environment="GARMINCONNECT_USERNAME=user"
Environment="GARMINCONNECT_PASSWORD=password"
ExecStart=gols --debug upload /media/fenix2/Garmin/Activity -m -c /home/user/
↪.config/gols/fit
[Install]
WantedBy=media-fenix2.mount
```

CHAPTER 4

# Reference

## gols

CHAPTER 5

# Contributing

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

## Bug reports

When reporting a bug please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

## Documentation improvements

gols could always use more documentation, whether as part of the official gols docs, in docstrings, or even on the web in blog posts, articles, and such.

## Feature requests and feedback

The best way to send feedback is to file an issue at https://github.com/euri10/gols/issues.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that code contributions are welcome :)

# Development

To set up *gols* for local development:

1. Fork gols (look for the "Fork" button).
2. Clone your fork locally:

```
git clone git@github.com:your_name_here/gols.git
```

3. Create a branch for local development:

```
git checkout -b name-of-your-bugfix-or-feature
```

   Now you can make your changes locally.

4. When you're done making changes, run all the checks, doc builder and spell checker with tox one command:

```
tox
```

5. Commit your changes and push your branch to GitHub:

```
git add .
git commit -m "Your detailed description of your changes."
git push origin name-of-your-bugfix-or-feature
```

6. Submit a pull request through the GitHub website.

## Pull Request Guidelines

If you need some code review or feedback while you're developing the code just make the pull request.

For merging, you should:

1. Include passing tests (run `tox`)[1].
2. Update documentation when there's new API, functionality etc.
3. Add a note to `CHANGELOG.rst` about the changes.
4. Add yourself to `AUTHORS.rst`.

## Tips

To run a subset of tests:

```
tox -e envname -- py.test -k test_myfeature
```

To run all the test environments in *parallel* (you need to `pip install detox`):

```
detox
```

[1] If you don't have all the necessary python versions available locally you can rely on Travis - it will run the tests for each change you add in the pull request.
It will be slower though ...

CHAPTER 6

# Authors

- Benoit Barthelet

CHAPTER 7

# Changelog

## 0.1.0 (2017-09-06)

- First release on PyPI.

CHAPTER 8

# Indices and tables

- genindex
- modindex
- search

# Python Module Index

## g

gols, 7

# Index

## G

gols (module), 7