

---

# **gitime Documentation**

*Release 1.0.1*

**Jonathan Goodnow**

November 17, 2015



<b>1</b>	<b>Installation</b>	<b>3</b>
<b>2</b>	<b>Getting Started</b>	<b>5</b>
<b>3</b>	<b>Command Details</b>	<b>7</b>
3.1	Invoices . . . . .	7
3.2	Commits . . . . .	8
3.3	Timer . . . . .	8
3.4	Status . . . . .	9
3.5	Export . . . . .	10
3.6	Settings . . . . .	11
3.7	Reset . . . . .	11



Gitime lets you keep track of your billable hours along with your commits. Build an invoice with your tasks and hours worked from your commit messages.



---

## Installation

---

Gitime can run on Linux, Windows, and Mac.

You'll only need two things installed to use gitime:

- [Git](#), and an executable called `git` on your path
- [Python 2.7](#) (or Python 3.4)

Install the latest gitime release with:

```
$ pip install gitime
```

Or install the development version with:

```
$ git clone https://github.com/jongoodnow/gitime
$ cd gitime
$ pip install -r requirements.txt
$ python setup.py install
```





---

## Getting Started

---

The first thing you should do is set your standard hourly rate, if you have one. You can manually set this for each invoice as well. To charge \$50 per hour, run:

```
$ gitime set -r 50
```

By default, gitime will round your hours worked to the nearest quarter hour. This can be changed to say, the nearest half-hour with:

```
$ gitime set --round 0.5
```

Now, create your first invoice, named after whatever project you're working on. `-r` and `--round` are available here too, if you want a custom rate for this project:

```
$ gitime invoice -n "Awesome Secret Project"
```

When you're ready to work, start the timer:

```
$ gitime timer start
```

You can pause, reset, or check the timer by replacing `start` with `pause`, `reset`, and `status` respectively.

When you're ready, make your commit as you would normally, but change the `git` to `gitime` on the commit step:

```
$ git add .  
$ gitime commit -m "Fixed a couple things"  
$ git push
```

Your time will be logged automatically, and the commit will be made. When you're ready for the next task, run `gitime timer start` again and repeat the process.

If you don't want to use the timer and would rather keep track of the time yourself, run your commit with the `--hours` flag:

```
$ gitime commit -m "Fixed a couple things" --hours 3
```

If you don't want to actually want to make a commit but want to log a task, run the commit with the `--fake` flag. Git will not be called.

You can check on your progress with:

```
$ gitime status
```

Or export your invoice to csv or xlsx with:

```
$ gitime export -f xlsx
```

---

## Command Details

---

### 3.1 Invoices

Switch to a different invoice, start a new one, or change invoice settings.

#### 3.1.1 Usage

```
gitime invoice [-h] [-n] [-r [RATE]] [-o [ROUNDING]] [-l] [name]
```

#### 3.1.2 Details

Invoices are used to store a log of your commit messages, commit dates, and hours worked.

#### 3.1.3 Arguments

##### name

The name of the invoice

#### 3.1.4 Options

##### **-new, -n**

Create a new invoice, and make it the active invoice where future commits are sent. Without this option, if you give a name that does not match any existing invoice, you will be asked if you want to make a new one.

##### **-rate [RATE], -r [RATE]**

Set the hourly rate in dollars to be used on the invoice specified by name.

##### **-round [ROUNDING], -rounding [ROUNDING], -o [ROUNDING]**

Choose how to round hours worked on the invoice specified by name. This defaults to the nearest quarter hour. For example, 2.71 hours will be rounded to 3 hours. If this setting is set to 0.5, 2.71 hours rounds to 2.5 hours.

**-list, -l**

Display a list of all your invoices.

**-help, -h**

Display a help message.

## 3.2 Commits

Run a regular git commit, but also log the commit message and time worked your active invoice.

### 3.2.1 Usage

```
gitime [-h] [--fake] [--hours [HOURS]] <your commit code ...>
```

### 3.2.2 Details

Your commit must contain a message in the form `-m [MSG]` or `--message=[MSG]`

### 3.2.3 Options

**-fake**

Your commit message and time will be logged in your active invoice, but no actual git commit will occur.

**-hours [HOURS]**

In lieu of using the timer, you can manually specify the amount of hours worked. The number you enter is subject to the rounding rules set by your invoice. By default, this will round the number entered here to the nearest quarter hour. If the timer has not tracked any time, this option is required. Otherwise, the timer values are used by default if this option is not specified.

**-help, -h**

Display a help message.

## 3.3 Timer

Start the timer, pause it, or reset it. The timer is reset automatically when you make a commit.

### 3.3.1 Usage

```
gitime timer [-h] [-f] {start,pause,reset,status}
```

### 3.3.2 Details

The timer is reset automatically when you make a commit. You will need to start the timer again after making a commit. Using the timer is optional. You can also use the `--hours` option when committing to specify the hours yourself.

### 3.3.3 Subcommands

#### start

Starts or restarts the timer, resuming from pause if necessary.

#### pause

Stops the timer, allowing you to restart with `start`.

#### reset

Stop the timer and set the time recorded to zero.

#### status

Show if the timer is running, and how much time it has recorded.

### 3.3.4 Options

#### `-force, -f`

The timer will not start if you do not have any invoices because it will have nowhere to record the time when you commit otherwise. If you want to start the timer anyway, use this flag to suppress that warning.

#### `-help, -h`

Display a help message.

## 3.4 Status

See what commits are in your current invoice.

### 3.4.1 Usage

```
gitime status [-h] [-i [INVOICE]]
```

### 3.4.2 Example

```
$ gitime status
Total Time Worked: 7.0 hours
Total Charges:      $350.00
Charges:
07-21-2014         2 hours      Fixed some css
07-21-2014         1 hours      Added a functor
07-22-2014         4 hours      Proved P!=NP
```

### 3.4.3 Options

**-invoice [INVOICE], -i [INVOICE]**

Choose an invoice by name, instead of using the active one.

**-help, -h**

Display a help message.

## 3.5 Export

Export your invoice.

### 3.5.1 Usage

```
gitime export [-h] [-i [INVOICE]] [-f [FORMAT]] [-p [FILE]]
```

### 3.5.2 Options

**-invoice [INVOICE], -i [INVOICE]**

The name of the invoice to export. Defaults to your active invoice. See what invoice is active with `gitime status`.

**-format [FORMAT], -f [FORMAT]**

Choose the export file format. This defaults to `csv`.

The formats currently available are:

- `csv`
- `xlsx`

**-file [FILE], -p [FILE]**

The name of the file to write. Defaults to the name of the invoice with an appropriate extension.

**-help, -h**

Display a help message.

## 3.6 Settings

Set up some default options.

### 3.6.1 Usage

```
gitime set [-h] [-r [RATE]] [-o [ROUNDING]] [-l]
```

### 3.6.2 Details

These options will change the defaults on invoices you create in the future. Existing invoices are not affected.

### 3.6.3 Options

**-rate [RATE], -r [RATE]**

Set the hourly rate in dollars to be used by default on all future invoices.

**-round [ROUNDING], -rounding [ROUNDING], -o [ROUNDING]**

Choose how to round hours worked. This defaults to the nearest quarter hour. For example, 2.71 hours will be rounded to 2.75 hours. If this setting is set to 0.5, 2.71 hours rounds to 2.5 hours.

**-list, -l**

Display what the `--rate` and `--round` settings are currently set to.

**-help, -h**

Display a help message.

## 3.7 Reset

Delete all your records and start afresh.

### 3.7.1 Usage

```
gitime reset [-h] [-f]
```

### 3.7.2 Details

This command will do the following:

- Erase all commit records on invoices (your actual git repo won't be affected)
- Erase all invoices
- Reset all default settings

Basically, gitime will be as it was when you first installed it.

**This cannot be undone.** You should export your invoices first.

### 3.7.3 Options

#### **-force, -f**

This option is for the bold. Without it, you will be asked if you are sure you want to do this first. With it, no checks will occur.

#### **-help, -h**

Display a help message.