PyTM Documentation

Release 0.0.16

Wasi

Nov 10, 2023

CONTENTS

PyTM

CHAPTER ONE

GOALS

Project time management, billing, and invoice generation.

CHAPTER

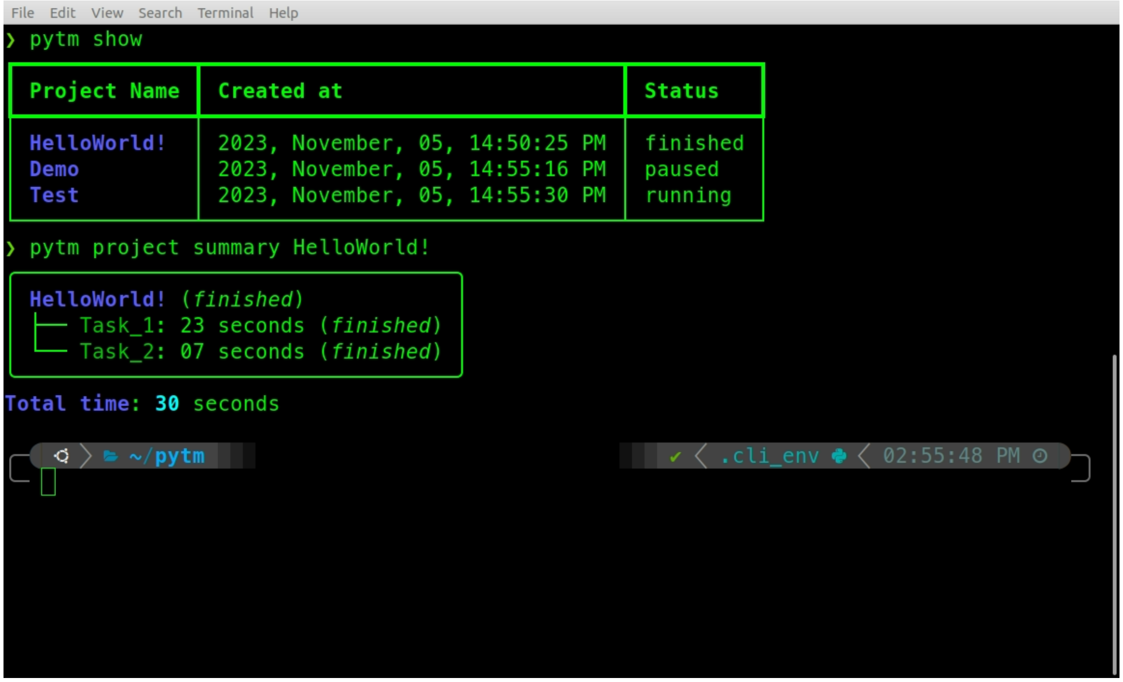
TWO

PREVIEW

CHAPTER

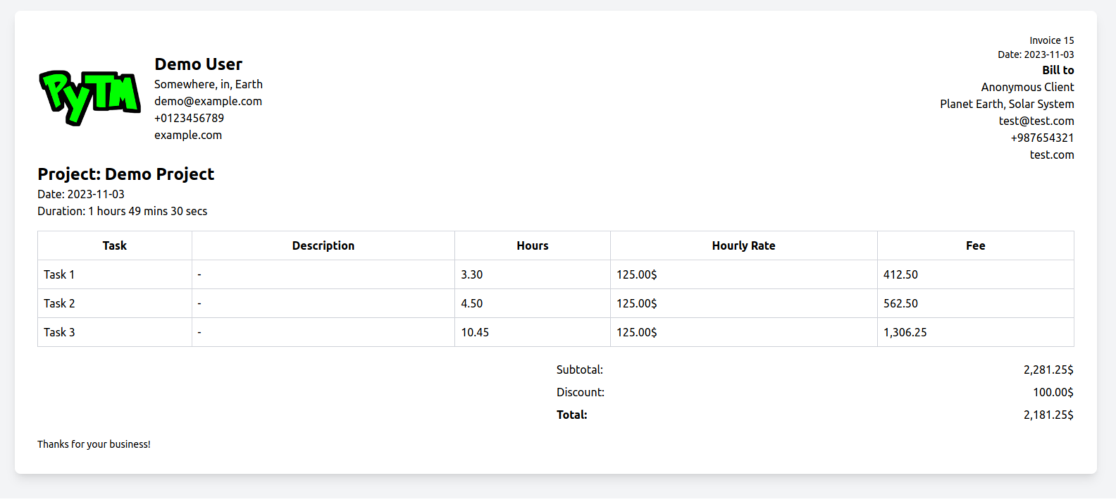
THREE

SCREENSHOTS



```
❯ pytm show
┌─────────────┬─────────────────────────────────┬──────────┐
│ Project Name│ Created at                      │ Status   │
├─────────────┼─────────────────────────────────┼──────────┤
│ HelloWorld! │ 2023, November, 05, 14:50:25 PM │ finished │
│ Demo        │ 2023, November, 05, 14:55:16 PM │ paused   │
│ Test        │ 2023, November, 05, 14:55:30 PM │ running  │
└─────────────┴─────────────────────────────────┴──────────┘
❯ pytm project summary HelloWorld!
┌───────────────────────────────────────┐
│ HelloWorld! (finished)                │
│ ├── Task_1: 23 seconds (finished)     │
│ └── Task_2: 07 seconds (finished)     │
└───────────────────────────────────────┘
Total time: 30 seconds

 ~/pytm                                    ✔ .cli_env  02:55:48 PM
```



Demo User
Somewhere, in, Earth
demo@example.com
+0123456789
example.com

Invoice 15
Date: 2023-11-03
Bill to
Anonymous Client
Planet Earth, Solar System
test@test.com
+987654321
test.com

Project: Demo Project
Date: 2023-11-03
Duration: 1 hours 49 mins 30 secs

Task	Description	Hours	Hourly Rate	Fee
Task 1	-	3.30	125.00$	412.50
Task 2	-	4.50	125.00$	562.50
Task 3	-	10.45	125.00$	1,306.25

Subtotal:	2,281.25$
Discount:	100.00$
Total:	2,181.25$

Thanks for your business!

CHAPTER

FOUR

INSTALLING PYTM

- First download and install pyenv. Use the command:

  ```
  curl https://pyenv.run | bash
  ```

- Next, install Python 3.12 using the command:

  ```
  pyenv install 3.12.0
  ```

  Alternatively, you can skip pyenv installation and download python 3.12 or above from the official website and setup a virtualenv as well.

- Next, install PyTM from PyPI using `pip`:

  ```
  python -m pip install python-pytm
  ```

Check the version by typing the following in your terminal.:

```
pytm --version
```

CHAPTER

FIVE

BASIC COMMANDS

To see the available commands type:

```
pytm --help
```

5.1 Commands related to projects

- Start a new project with a default name: `pytm project start`
- Start a new project with the given name or, start an existing project: `pytm project start PROJECT_NAME`
- Rename a project: `pytm project rename OLD_PROJECT_NAME NEW_NAME`
- Remove a project: `pytm project remove PROJECT_NAME`
- Check the status of a project: `pytm project status PROJECT_NAME`
- Check the list of tasks and duration of a project: `pytm project summary PROJECT_NAME`
- Finish active project: `pytm project finish`
- Pause active project: `pytm project pause`
- Abort active project: `pytm project abort`

5.2 Commands related to Task

- Start a new task with a default name in the current active project: `pytm task start`
- Start a new task with the given name or existing task in the current active project: `pytm task start TASK_NAME`
- Rename a task of the active project: `pytm task rename OLD_TASK_NAME NEW_NAME`
- Remove a task: `pytm task remove TASK_NAME`
- current task's status: `pytm task status`
- Finish active task: `pytm task finish`
- Pause active task: `pytm task pause`
- Abort active task: `pytm task abort`

5.3 Others

Configure project, user and invoice info:

```
pytm config project PROJECT_NAME
pytm config user
pytm config invoice
```

Generate Invoice:

```
pytm invoice auto PROJECT_NAME
pytm invoice manual
```

Check version:

```
pytm --version
pytm -v
```

Check summary of all the projects:

```
pytm summary
```

For a list of all the available commands try:

```
pytm --help
```

CHAPTER

SIX

RUNNING THE TESTS

- Clone this repository
- Install dependencies:

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

- run the tests:

```
py.test
```

CHAPTER

SEVEN

NOTES

- **Author** - Wasi - (wasi0013).
- **License** - see the LICENSE file.
- **Contributing** - see CONTRIBUTING.rst for detail. You can also help by creating issues.
- **Version** - see the tags on this repository.
- **Acknowledgments** - bootstrapped using this cookiecutter package.
- Built With :heart: using Python.

7.1 Contents:

7.1.1 Installing PyTM

You can download it from Python Package Index! For example:

```
pip install python-pytm
```

or, you can install it using pipenv too!:

```
pipenv install python-pytm
```

Checkout the version by typing the following in your terminal/command prompt:

```
pytm --version
```

For a list of all the available options or, arguments try:

```
pytm --help
```

7.1.2 Usage

For installation information Checkout: this instruction

Basic commands:

Currently available commands are listed below:

Commands related to project

- Start a new/existing project:

  ```
  pytm project start PROJECT_NAME
  ```

- Remove a project:

  ```
  pytm project remove PROJECT_NAME
  ```

- Check status of a project:

  ```
  pytm project status PROJECT_NAME
  ```

- Check summary of a project:

  ```
  pytm project summary PROJECT_NAME
  ```

- Finish active project:

  ```
  pytm project finish
  ```

- Pause active project:

  ```
  pytm project pause
  ```

- Abort active project:

  ```
  pytm project abort
  ```

Commands related to Task

- Start a new or existing task in the current active project:

  ```
  pytm task start TASK_NAME
  ```

- Remove a task:

  ```
  pytm task remove TASK_NAME
  ```

- current task’s status:

  ```
  pytm task status
  ```

- Finish active task:

```
pytm task finish
```

- Pause active task:

```
pytm task pause
```

- Abort active task:

```
pytm task abort
```

Others

Check version:

```
pytm --version
pytm -v
```

For a list of all the available options or, arguments try:

```
pytm --help
```

7.1.3 Contributing

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

You can contribute in many ways:

Types of Contributions

Report Bugs

Report bugs at https://github.com/wasi0013/PyTM/issues.

If you are 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.

Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” is open to whoever wants to implement it.

Implement Features

Look through the GitHub issues for features. Anything tagged with “feature” is open to whoever wants to implement it.

Write Documentation

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

Submit Feedback

The best way to send feedback is to file an issue at https://github.com/wasi0013/PyTM/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 contributions are welcome :)

Get Started!

Ready to contribute? Here’s how to set up *PyTM* for local development.

1. Fork the *PyTM* repo on GitHub.
2. Clone your fork locally:

   ```
   $ git clone git@github.com:your_name_here/PyTM.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, check that your changes pass style and unit tests, including testing other Python versions with tox:

   ```
   $ tox
   ```

To get tox, just pip install it.

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

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 2.6, 2.7, and 3.3, and for PyPy. Check https://travis-ci.org/wasi0013/PyTM under pull requests for active pull requests or run the `tox` command and make sure that the tests pass for all supported Python versions.

Tips

To run a subset of tests:

```
$ py.test test/test_PyTM.py
```

7.1.4 Credits

Development Lead

- *Wasi <https://hellowasi.com>*

Contributors

- *Ashutosh <https://github.com/pyprism>*

7.1.5 History

7.1.6 0.0.1 (2018-10-23)

- First release on PyPI.

7.1.7 0.0.2 (2018-11-09)

- Second release on PyPI.
- Created Skeleton of the command line Interface.

7.1.8 0.0.3 (2023-10-18)

- Implemented data store in home directory.
- Made project commands functional.

7.1.9 0.0.4 (2023-10-28)

- Implemented state.
- Improved project commands.

7.1.10 0.0.5 (2023-10-30)

- Made task commands functional.
- Improved project summary.
- Project summary now shows total duration of the project.

7.1.11 0.0.6 (2023-10-31)

- bug fix
- readme and doc update
- Improved Command line interface outputs.

7.1.12 0.0.7 (2023-11-01)

- Added show sub command for project.
- Added json sub command for project.
- refactored project and task commands output messages.
- fixed bugs in project sub commands.

7.1.13 0.0.8 (2023-11-01)

- Added config sub command.
- added command to save default user information.
- refactored project and task commands.
- fixed bugs.
- added config command to configure project meta data such as title, billable? etc.

7.1.14 0.0.9 (2023-11-02)

- Added config invoice command for configuring default invoice texts and logo.
- Added invoice sub command.
- Added invoice manual command to generate manual invoice using prompts.
- Added invoice auto <Project> command to generate invoice from existing projects.
- invoice manual and auto command generates HTML invoices using Tailwind CSS and opens it in a Browser.

7.1.15 0.0.10 (2023-11-03)

- open invoices using python's standard library.
- code refactor
- bug fix

7.1.16 0.0.11 (2023-11-03)

- Major bug fix.

7.1.17 0.0.12 (2023-11-04)

- Bug fix.

7.1.18 0.0.13 (2023-11-07)

- Made the init command optional.

7.1.19 0.0.14 (2023-11-07)

- config invoice command bug fix.
- copy logo instead of moving it permanently when configuring invoice.
- fixed f strings quote issue.
- made init command hidden.
- improved doc strings and help texts.

7.1.20 0.0.15 (2023-11-08)

- added new command rename for renaming existing projects.
- added new command rename for renaming existing tasks.
- updated show command to show active task and project when available.
- Simplified project start command.
- New projects can be started without project name.
- Project started without a name will now have a default name which can be renamed later.
- Simplified task start command.
- New tasks can be started without task name.
- Task started without a name will have a default name which can be renamed later at ease.
- Bug fix

7.1.21 0.0.16 (2023-11-11)

- added new command summary.
- fixed bug and refactored invoice handler.
- added tests for project handler.
- added tests for task handler.
- added tests for invoice handler.
- updated make command to update README coverage percentage while testing.

7.2 Feedback

If you have any suggestions or questions about **PyTM** feel free to email me at wasi0013@gmail.com.

If you encounter any errors or problems with **PyTM**, please let me know! Open an Issue at the GitHub http://github.com/wasi0013/PyTM main repository.