
Parasut CLI Documentation

Release 0.6.2

Metehan Gulac

Apr 08, 2022

CONTENTS:

1	Parasut CLI	1
1.1	Features	1
1.2	TODO	1
2	Installation	3
2.1	Dependency Setup	3
2.2	Workspace Structure	4
2.3	Environment Variables	4
2.4	Stable release	5
2.5	From sources	5
3	Usage	7
3.1	Start Command	7
3.2	Link Command	7
3.3	Switch Command	8
3.4	Run Command	8
3.5	Release Command	8
3.6	Version Command	9
3.7	Help Command	9
4	API Reference	11
4.1	parasut_cli package	11
5	Contributing	13
5.1	Types of Contributions	13
5.2	Get Started!	14
5.3	Pull Request Guidelines	15
5.4	Tips	15
5.5	Deploying	15
6	Credits	17
6.1	Development Lead	17
6.2	Contributors	17
7	History	19
7.1	0.6.1 (2022-01-05)	19
7.2	0.6.0 (2022-01-05)	19
7.3	0.5.0 (2022-01-04)	19
7.4	0.4.9 (2021-11-17)	19
7.5	0.4.8 (2021-10-12)	19
7.6	0.4.7 (2021-09-02)	20

7.7	0.4.6 (2021-08-26)	20
7.8	0.4.5 (2021-08-13)	20
7.9	0.4.4 (2021-08-13)	20
7.10	0.4.3 (2021-08-13)	20
7.11	0.4.2 (2021-08-12)	20
7.12	0.4.1 (2021-05-06)	20
7.13	0.4.0 (2021-05-04)	21
7.14	0.3.5 (2021-03-23)	21
7.15	0.3.4 (2021-03-23)	21
7.16	0.3.3 (2021-03-19)	21
7.17	0.3.2 (2021-03-19)	21
7.18	0.3.1 (2021-03-19)	21
7.19	0.3.0 (2021-03-19)	22
7.20	0.2.0 (2021-03-17)	22
7.21	0.1.16 (2021-03-09)	22
7.22	0.1.15 (2021-03-09)	22
7.23	0.1.14 (2021-03-08)	22
7.24	0.1.13 (2021-03-08)	23
7.25	0.1.12 (2021-03-08)	23
7.26	0.1.11 (2021-03-08)	23
7.27	0.1.10 (2021-03-07)	23
7.28	0.1.9 (2021-03-07)	23
7.29	0.1.8 (2021-03-07)	23
8	FAQ	25
8.1	How to upgrade current version	25
9	Indices and tables	27
	Python Module Index	29
	Index	31

PARASUT CLI

Parasut development environment consists of microservices and different applications depending on them. **parasut-cli** is a CLI that facilitates you to manage your workspaces and the applications you will run during development.

- Free software: MIT license
- Documentation: <https://parasut-cli.readthedocs.io>.

1.1 Features

- **start** command for preparing workspace with all necessary options.
- **link** command for yarn linking operations. Also **–undo** action.
- **switch** command for rails console actions.
- **run** command for executing repo command chains manually.
- **release** command for releasing version on related repo.
- **pre-release** flag for **release** command.
- **rich** integrated.

1.2 TODO

- **clone** command for cloning repos in base directory.
- unit tests.
- mypy & black connection to tox.
- linking state mechanism refactor.

INSTALLATION

2.1 Dependency Setup

There are some dependencies we need to handle before running the cli.

It is necessary to install and configure these third party tools below:

- [Ember.js](#)
- [Ruby on Rails](#)
- [asdf](#) - version manager
- [tmux](#) - terminal multiplexer
- [npm-cli-login](#) - npm auto login package (optional)

Warning: Before using the asdf version manager, you need to install the interpreters and their valid versions you will use (maven, nodejs, ruby, yarn):

```
$ asdf plugin-add maven
$ asdf install maven <version>
$ asdf plugin-add nodejs
$ asdf install nodejs <version>
$ asdf plugin-add ruby
$ asdf install ruby <version>
$ asdf plugin-add yarn
$ asdf install yarn <version>
```

Note: **asdf** will create a file named `.tool-versions` in all your repositories while CLI is running. To have Git ignore these files, you can create a global `.gitignore` file and define this information in git.

```
$ echo .tool-versions > ~/.gitignore
$ git config --global core.excludesFile '~/.gitignore'
```

2.2 Workspace Structure

Clone all repos into a folder that you will use as base. You get the idea.:

```
parasutcom/      # Parasut Base Directory
  server/        # Server repo
  billing/       # Billing repo
  e-doc-broker/  # broker repo
  post-office/   # post-office repo
  printX/        # printX repo
  ubl-validator/ # ubl-validator repo
  phoenix/       # phoenix repo
  shared-logic/  # shared repo
  trinity/       # trinity repo
  ui-library/    # ui-library repo
```

2.3 Environment Variables

CLI only need these certain `environment` variables while working. You need to define them in your local shell configuration. As long as you pay attention here, everything goes well on CLI. Briefly define the following variables:

```
# parasut-cli text editor
export PARASUT_CLI_TEXT_EDITOR="vim"
# company id
export PARASUT_COMPANY_ID="..."
# npm registry
export PARASUT_REGISTRY="..."
export PARASUT_NPM_USERNAME="..."
export PARASUT_NPM_PASSWORD="..."
export PARASUT_NPM_EMAIL="..."
# switch rails names
export PARASUT_PHOENIX_SWITCH_NAME="..."
export PARASUT_PHOENIX_SWITCH_PRICING_LIST_NAME="..."
export PARASUT_TRINITY_SWITCH_NAME="..."
export PARASUT_TRINITY_SWITCH_PRICING_LIST_NAME="..."
# version & ports
export PARASUT_SERVER_RUBY_V="..."
export PARASUT_BILLING_RUBY_V="..."
export PARASUT_BILLING_RAILS_PORT="..."
export PARASUT_E_DOC_BROKER_RUBY_V="..."
export PARASUT_E_DOC_BROKER_RAILS_PORT="..."
export PARASUT_POST_OFFICE_RUBY_V="..."
export PARASUT_POST_OFFICE_RAILS_PORT="..."
export PARASUT_UBL_VALIDATOR_MAVEN_V="..."
export PARASUT_PHOENIX_NODE_V="..."
export PARASUT_PHOENIX_YARN_V="..."
export PARASUT_CLIENT_NODE_V="..."
export PARASUT_CLIENT_YARN_V="..."
export PARASUT_CLIENT_EMBER_PORT="..."
export PARASUT_TRINITY_NODE_V="..."
export PARASUT_TRINITY_YARN_V="..."
```

(continues on next page)

(continued from previous page)

```

export PARASUT_TRINITY_EMBER_PORT="..."
export PARASUT_UI_LIBRARY_NODE_V="..."
export PARASUT_UI_LIBRARY_YARN_V="..."
export PARASUT_UI_LIBRARY_EMBER_PORT="..."
export PARASUT_SHARED_LOGIC_NODE_V="..."
export PARASUT_SHARED_LOGIC_YARN_V="..."
export PARASUT_SHARED_LOGIC_EMBER_PORT="..."
export PARASUT_PRINTX_NODE_V="..."
export PARASUT_PRINTX_YARN_V="..."
export PARASUT_PRINTX_EMBER_PORT="..."
# structure variables
export PARASUT_BASE_DIR=~/.Code/development/parasutcom"
export PARASUT_SERVER_DIR="server"
export PARASUT_BILLING_DIR="billing"
export PARASUT_E_DOC_BROKER_DIR="e-doc-broker"
export PARASUT_POST_OFFICE_DIR="post-office"
export PARASUT_UBL_VALIDATOR_DIR="ubl-validator"
export PARASUT_PHOENIX_DIR="phoenix"
export PARASUT_SHARED_LOGIC_DIR="shared-logic"
export PARASUT_PRINTX_DIR="printX"
export PARASUT_CLIENT_DIR="client"
export PARASUT_TRINITY_DIR="trinity"
export PARASUT_UI_LIBRARY_DIR="ui-library"

```

Warning: CLI using text editor like vim, nvim, emacs, nano while executing `start` command for launching tmux server. It will try to open editor in tmux window. Don't use anything else. If you're using IDE or something else that work outside of terminal, just skip `-e/--edit` partition of `start` command.

2.4 Stable release

To install Parasut CLI, run this command in your terminal:

```
$ pip install parasut-cli
```

This is the preferred method to install Parasut CLI, as it will always install the most recent stable release.

If you don't have `pip` installed, this [Python installation guide](#) can guide you through the process.

2.5 From sources

The sources for Parasut CLI can be downloaded from the [Github repo](#).

You can either clone the public repository:

```
$ git clone git://github.com/mthnglac/parasut-cli
```

Or download the [tarball](#):

```
$ curl -OJL https://github.com/mthnglac/parasut-cli/tarball/master
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```

3.1 Start Command

This is where the CLI does its magic. It creates two sessions from repos given using tmux and builds workspaces into them.

```
$ parasut-cli start -e <repo-name> -s <repo-name>
```

- **-e/--edit** - a repository name to open in text editor. Choices: server, billing, phoenix, shared-logic, trinity, ui-library, client, e-doc-broker, post-office, ubl-validator, printX.
- **-s/--setup** - a repository name to launch. Choices: server, billing, phoenix, shared-logic, trinity, ui-library, client, e-doc-broker, post-office, ubl-validator, printX.
- **-w/--worker** - a worker name to launch. This is necessary when you working with e-invoice. Choices: server-worker, e-doc-broker-worker.

Note: You can access existing tmux sessions with this command.

```
$ tmux list-sessions  
$ tmux a -t <session_name>
```

3.2 Link Command

This command does the linking work of yarn. It takes the given argument and writes it to package.json in the current repo for you. It also includes a logic so that you can undo any changes you have made later (**-u/--undo**).

```
$ parasut-cli link -b <repo-name> -t <repo-name>  
$ parasut-cli link -b <repo-name> -t <repo-name> --output  
$ parasut-cli link -b <repo-name> -t <repo-name> <repo-name> --output
```

- **-b/--base** - a repository name for linking target repository. Use this with **-t/--target** option. Choices: phoenix, trinity.
- **-t/--target** - a target repository name for linking it to base repository. Choices: ui-library, shared-logic.
- **-u/--undo** - a repository name for unlinking. Choices: ui-library, shared-logic.
- **--list** - a repository name for unlinking it list linked repos of base repo.
- **--output** - an option for showing process output.

Note: -b/--base argument is necessary. Command will continue to ask until the required base repo is given.

3.3 Switch Command

This part can be a little confusing. It represents exactly what it does. Switch processes run on **server** and **billing**. What the command does is to automate processes that are executed interactively in the background.

```
$ parasut-cli switch <switch-name> -t <switch-choice>
```

- **frontend** - a repository name to switch frontend repo on server.
 - -t/--target - target prefix for choosing choices. Choices: phoenix, trinity.
 - --output - an option for showing process output.
- **addlings** - command for switching addlings.
 - -t/--target - target prefix for choosing choices. Choices: receipt, invoice.
 - --output - an option for showing process output.
- **pricing_list** - command for switching addlings.
 - -t/--target - target prefix for choosing choices. Choices: trinity, phoenix.
 - --output - an option for showing process output.

3.4 Run Command

This command runs the chain of commands required to instantiate the repository. Selects yarn, node and ruby versions and related options for the target repo.

```
$ parasut-cli run -t <repo-name>
```

- -t/--target - a repository name for running target repository with necessary options. Choices: server, server-sidekiq, billing, billing-sidekiq, e-doc-broker, e-doc-broker-sidekiq, post-office, post-office-sidekiq, ubl-validator, phoenix, shared-logic, trinity, ui-library, client, printX.

3.5 Release Command

It does the “release” operation on the target repo. It does simply sets the registry, logs in to the npm, publishes the package, and deletes the registry.

```
$ parasut-cli release -t <repo-name>
$ parasut-cli release -t <repo-name> --output
$ parasut-cli release -t <repo-name> --pre-release --output
```

- -t/--target - a repository name for publish release to target repository. Choices: shared-logic, ui-library.
- --pre-release - an option for making pre-release from your local branch (beta).
- --auto-login - an option for logging in to npm automatically.

- `--output` - an option for showing process output.

Note: You have to set these environment variables be able to release the target repo:

```
$ export PARASUT_NPM_USERNAME=<npm_username>
$ export PARASUT_NPM_PASSWORD=<npm_password>
$ export PARASUT_NPM_EMAIL=<npm_email>
```

Also, if you want to login to npm without entering username and password, you are lucky because parasut-cli has a flag for this: `--auto-login`. But first, you have to install the third party package below; after the installation you can use `--auto-login` flag.

```
$ npm install -g npm-cli-login
$ parasut-cli release -t <repo-name> --auto-login
$ parasut-cli release -t <repo-name> --auto-login --output
$ parasut-cli release -t <repo-name> --pre-release --auto-login --output
```

3.6 Version Command

You can use this command to show the version of the CLI.

```
$ parasut-cli -v
$ parasut-cli --version
```

3.7 Help Command

Generally, the CLI will not make you do anything against the rules and will give you the necessary warnings when the time comes. If you are confused, type `-h/--help` at the end of your chain of commands:

```
$ parasut-cli -h
$ parasut-cli start --help
$ parasut-cli start -b <repo-name> --help
$ parasut-cli switch --help
```


API REFERENCE

4.1 parasut_cli package

Top-level package for Parasut CLI.

4.1.1 Subpackages

parasut_cli.commands package

Submodules

parasut_cli.commands.link module

parasut_cli.commands.start module

parasut_cli.commands.switch module

Module contents

parasut_cli.config package

Submodules

parasut_cli.config.settings module

Module contents

parasut_cli.utils package

Submodules

parasut_cli.utils.command module

class parasut_cli.utils.command.Command

Bases: abc.ABC

abstract execute() → None

parasut_cli.utils.invoker module

class parasut_cli.utils.invoker.**Invoker**

Bases: object

do_something_important(*command*: parasut_cli.utils.command.Command) → None

set_on_finish(*command*: parasut_cli.utils.command.Command) → None

set_on_start(*command*: parasut_cli.utils.command.Command) → None

parasut_cli.utils.receiver module

Module contents

4.1.2 Submodules

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:

5.1 Types of Contributions

5.1.1 Report Bugs

Report bugs at <https://github.com/mthnglac/parasut-cli/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.

5.1.2 Fix Bugs

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

5.1.3 Implement Features

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

5.1.4 Write Documentation

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

5.1.5 Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/mthnglac/parasut-cli/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 :)

5.2 Get Started!

Ready to contribute? Here's how to set up *parasut-cli* for local development.

1. Fork the *parasut-cli* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:mthnglac/parasut-cli.git
```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv parasut-cli
$ cd parasut-cli/
$ python setup.py develop
```

4. Create a branch for local development:

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

Now you can make your changes locally.

5. Check typings with mypy. It will check all relative files through cli.py:

```
$ mypy parasut_cli/cli.py
```

To get mypy, just pip install them into your virtualenv.

6. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
$ flake8 parasut-cli tests
$ python setup.py test or pytest
$ tox
```

To get flake8 and tox, just pip install them into your virtualenv.

7. 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
```

8. Submit a pull request through the GitHub website.

5.3 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 3.6, 3.7, 3.8 and 3.9, and for PyPy. Check https://travis-ci.com/mthnglac/parasut-cli/pull_requests and make sure that the tests pass for all supported Python versions.

5.4 Tips

To run a subset of tests:

```
$ python -m unittest tests.test_parasut_cli
```

5.5 Deploying

A reminder for the maintainers on how to deploy. Make sure all your changes are committed (including an entry in HISTORY.rst). Then run:

```
$ bump2version patch # possible: major / minor / patch
$ git push
$ git push --tags
```

Travis will then deploy to PyPI if tests pass.

CREDITS

6.1 Development Lead

- Metehan Gulac <metehanglc@protonmail.com>

6.2 Contributors

None yet. Why not be the first?

HISTORY

7.1 0.6.1 (2022-01-05)

- switch option for Asist.

7.2 0.6.0 (2022-01-05)

- FIX: --pre-release flag was broken. Also now flow was tested.

7.3 0.5.0 (2022-01-04)

- FEATURE: --pre-release flag added to release command.
- UPDATE: docs improvements.

7.4 0.4.9 (2021-11-17)

- UPDATE: add logic circle to switching frontend.
- PARASUT_PHOENIX_SWITCH_NAME environment variable changed as PARASUT_PHOENIX_SWITCH_APP_NAME
- PARASUT_PHOENIX_SWITCH_OWNER_TYPE_NAME environment variable added.
- UPDATE: minor changes on linking

7.5 0.4.8 (2021-10-12)

- DELETE: `git push --tags` command removed from release process; it is already done in “ember release” command.
- FEATURE: printX repo & its related logic & environments added.
- UPDATE: docs improvements

7.6 0.4.7 (2021-09-02)

- UPDATE: “git push ..” after release.

7.7 0.4.6 (2021-08-26)

- FEATURE: new switch subcommand added: `pricing_list`.
- UPDATE: docs improvements.
- UPDATE: minor bugs.

7.8 0.4.5 (2021-08-13)

- FEATURE: `--yes` flag added for ember release auto login.

7.9 0.4.4 (2021-08-13)

- BUG: auto release was working as manuel.
- UPDATE: docs improvements

7.10 0.4.3 (2021-08-13)

- REFACTOR: release command logic
- UPDATE: `--auto-login` and `--output` options for release command
- UPDATE: docs improvements

7.11 0.4.2 (2021-08-12)

- UPDATE: new core commands for releasing

7.12 0.4.1 (2021-05-06)

- `version` command added.
- bug fixings.
- docs improvements.

7.13 0.4.0 (2021-05-04)

- `post-office` repo added.
- `ubl-validator` repo added.
- `release` command added (beta).
- `tox python3.5` deprecated
- dependency upgrades.
- editor workspace choosing node,ruby versions now in favor of editor actions.
- all version managers (`rvm`, `yvm`, `nvm`) deprecated in favor of `asdf` .
- docs improvements.

7.14 0.3.5 (2021-03-23)

- UPDATE: rich integrated install script.
- UPDATE: docs improvements.

7.15 0.3.4 (2021-03-23)

- FIX: `run_process` output fix.

7.16 0.3.3 (2021-03-19)

- FIX: flake8 warnings.
- FIX: black warnings.

7.17 0.3.2 (2021-03-19)

- UPDATE: Docs improvements

7.18 0.3.1 (2021-03-19)

- UPDATE: Docs improvements
- DELETE: forgotten paremeters.

7.19 0.3.0 (2021-03-19)

- FEATURE: `rich` integrated. Plugin used in hidden commands.
- FEATURE: new `--output` parameter. `switch` and `link` command outputs are now hidden. To show their outputs, use `--output` prefix at the end of command chain.
- FIX: Yarn error situation was not working properly. Exception was not working. Also, CLI now gets angry for the third time.
- REFACTOR: subprocess structure has been rewritten. Ineractive terminal mode deprecated.

7.20 0.2.0 (2021-03-17)

- CREATE: API Reference page added.
- CREATE: `run` command added.
- UPDATE: pkg updates.

7.21 0.1.16 (2021-03-09)

- FIX: tmux exception problem when there is no server.

7.22 0.1.15 (2021-03-09)

- REFACTOR: `rails` subcommand changed as `frontend`.
- FIX: typos in core.
- FIX: dependencies now installing at installation.
- FIX: state mechanism refactored. There was a `FileNotFoundError`.
- FIX: `start` command now appending new windows existing session if you run `start` command in detached mode.
- UPDATE: note added to `start` command.

7.23 0.1.14 (2021-03-08)

- Docs improvements

7.24 0.1.13 (2021-03-08)

- Docs improvements

7.25 0.1.12 (2021-03-08)

- Docs improvements

7.26 0.1.11 (2021-03-08)

- Docs installation and usage page improvements.
- Sphinx upgrade.

7.27 0.1.10 (2021-03-07)

- Docs improvements

7.28 0.1.9 (2021-03-07)

- Documents added on installation.
- Some minor fixings.
- Static type checker mypy added to requirements.
- Black code formatter used.

7.29 0.1.8 (2021-03-07)

- Theme changed
- Switch command refactored. added options: addlings, rails.
- Link command base repo argument required now.
- All important environments now coming outside of project.
- Cli now checking env variables; at exception.Keyerror situation, cli logging missing argument with warning.
- Start command refactored.
- Link list command shortening removed. at the moment only option is “-list”.
- Parser indent fixed.

8.1 How to upgrade current version

```
pip install --upgrade parasut-cli
```


INDICES AND TABLES

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

PYTHON MODULE INDEX

p

- `parasut_cli`, [11](#)
- `parasut_cli.commands`, [11](#)
- `parasut_cli.config`, [11](#)
- `parasut_cli.utils`, [12](#)
- `parasut_cli.utils.command`, [11](#)
- `parasut_cli.utils.invoker`, [12](#)

INDEX

C

`Command` (class in `parasut_cli.utils.command`), [11](#)

D

`do_something_important()` (`parasut_cli.utils.invoker.Invoker` method), [12](#)

E

`execute()` (`parasut_cli.utils.command.Command` method), [11](#)

I

`Invoker` (class in `parasut_cli.utils.invoker`), [12](#)

M

module

- `parasut_cli`, [11](#)
- `parasut_cli.commands`, [11](#)
- `parasut_cli.config`, [11](#)
- `parasut_cli.config.settings`, [11](#)
- `parasut_cli.utils`, [12](#)
- `parasut_cli.utils.command`, [11](#)
- `parasut_cli.utils.invoker`, [12](#)

P

- `parasut_cli`
 - module, [11](#)
- `parasut_cli.commands`
 - module, [11](#)
- `parasut_cli.config`
 - module, [11](#)
- `parasut_cli.config.settings`
 - module, [11](#)
- `parasut_cli.utils`
 - module, [12](#)
- `parasut_cli.utils.command`
 - module, [11](#)
- `parasut_cli.utils.invoker`
 - module, [12](#)

S

`set_on_finish()` (`parasut_cli.utils.invoker.Invoker` method), [12](#)

`set_on_start()` (`parasut_cli.utils.invoker.Invoker` method), [12](#)