
nitpicker Documentation

Release 0.4.0

Kenneth Reitz

Oct 30, 2018

Contents

1	Nitpicker	1
1.1	Motivation	1
1.2	How does it work?	1
1.3	Features	2
1.4	Installation	2
1.5	Usage	3
1.6	Documentation	3

Nitpicker is a CLI tool for black-box testing written in Python

1.1 Motivation

The project has been started to fix some problems that many developers and testers, who practice black-box testing, might be familiar to:

1. Black-box testing are not under version control with the code. Why not? As developers, we would like to do some review of test cases like code review. As a manager I would be calm knowing that all the test plans and cases are stored with the code on Git repository and are always available.
2. Black-box testing stays apart from the develop cycle. I can ban a merge request if it breaks my unit or integration tests because I see it at once by using CI tools. I believe it is possible for manual tests too. I want my CI tool to check if a tester do all the needed tests.
3. A testing tool should be interactive. When you see a whole test case with all the steps it is hard not to jump between them trying to do test as fast as possible. When a tester is in a dialogue with a tool and goes step-by-step, they can test more carefully. Especially, if the tool keep time tracking automatically.

1.2 How does it work?

All your test cases and run reports are stored in YAML format with the code which they test.

```
project
|-src/
|-docs/
|-qa/
  |-feature_1/
  |-feature_2/
  |-plan_1/
```

(continues on next page)

(continued from previous page)

```
| -test_case1.yml
| -test_case2.yml
| -test_case3.yml
| -runs/
|   |-20180820_232000_run.report
|   |-20180820_232010_run.report
```

Nitpicker provides command to create a test case in the given test plan:

```
python -m nitpicker add test_case -p feature_1.plan_1
```

Then you should write the case by using your favourite text editor. It is a not bad idea to commit and push it, so your teammate can review the case before you run the plan which the case belongs to.

Now you can run the test plan:

```
python -m nitpicker run feature_1.plan_1
```

The program runs all the cases in the interactive mode leading the tester step by step. The results of the run will be written in directory *runs* in YAML format.

After all the test cases have been run you can push the reports into the git repo, so your CI server can check if all the test runs are passed

```
python -m nitpicker check --all-runs-passed
```

The project uses itself for testing. You can find *qa* directory in the repo. Also you can run some plans for demonstration.

1.3 Features

- CLI interface to create, run, validate and check user tests
- Run tests in an interactive mode
- Storing user tests and run reports as files in YAML format
- Integration with version control systems (currently only **git**)

1.4 Installation

```
pip install nitpicker
```

or

```
python -m pip install nitpicker
```

Currently Nitpicker supports Python 3.3 and newer.

If you are a Windows user and you would like to use Nitpicker as a command (run *nitpicker* instead of *python -m nitpicker*) you should add Script directory of your Python to PATH variable.

1.5 Usage

In order to start:

```
python -m nitpicker --help
```

1.6 Documentation

See the last documentation [here](#).