TIERRA
|TELEMATICS DESIGN|

cookiecutter-qa Documentation

Release 0.0.1

Audrey Roy Greenfeld

Apr 04, 2018

Contents

CHAPTER 1

Getting Started

1.1 cookiecutter QA

Coookiecutter QA let you create QA projects based on the Cookiecutter scaffolding project.

1.1.1 Usage

Install Cookiecutter, use `cookiecutter-qa` as project template and bake your new QA package providing some information:

```
$ pip install cookiecutter
$ cookiecutter https://github.com/tierratelematics/cookiecutter-qa
full_name [Davide Moro]:
email [davide.moro@gmail.com]:
github_username [tierratelematics]:
project_name [Project QA]:
project_slug [project_qa]:
project_short_description [Project QA contains all the boilerplate you need to create␣
↪a QA package]:
version [0.0.1]:
create_author_file [y]:
Select open_source_license:
1 - MIT license
2 - BSD license
3 - ISC license
4 - Apache Software License 2.0
5 - GNU General Public License v3
6 - Not open source
Choose from 1, 2, 3, 4, 5, 6 [1]: 4
Select command_line_interface:
1 - Click
2 - No command-line interface
Choose from 1, 2 [1]: 2
```

(continues on next page)

(continued from previous page)

```
base_url [https://www.tierratelematics.com]:
selenium_grid_url [http://USERNAME:ACCESS_KEY@hub.browserstack.com:80/wd/hub]: YOUR_
↪SELENIUM_GRID_URL_HERE
testrail [y]:
$ cd project_qa
```

As result cookiecutter will create for you a new package with a hello world test pytest, Splinter, BDD and page objects ready.

Important note: be aware that the *selenium_grid_url* will be saved in `project_name/Dockerfile` so keep in mind that before distributing your project!

1.1.2 How to use it

If you want to perform a quick tour create a BrowserStack free account and you will be able to run your tests against a real remote browser without having to install locally all the needed prerequisites (geckodriver, chromedriver, adjust executable paths, etc).

Once logged in on BrowserStack visit `Account > Settings`, copy the Automate's username and access key and generate a new cookiecutter project providing the remote selenium grid url following the format:

```
http://USERNAME:ACCESS_KEY@hub.browserstack.com:80/wd/hub
```

You can use any Selenium grid provider (SauceLabs, BrowserStack, TestingBot) or using your own local grid with Zalenium.

Docker

If you want to launch your hello world Selenium based tests against BrowserStack you can just type the following commands (Docker required):

```
$ make docker-run
```

or:

```
$ docker run --rm -it project_qa -epy36 -- \
    -vvv --splinter-webdriver=remote \
    --variables=credentials/credentials_template.yml \
    --splinter-remote-url=http://USERNAME:ACCESS_KEY@hub.browserstack.com:80/wd/hub \
    --variables capabilities/os/WIN10.json
    --variables capabilities/browsers/chrome/CHROME.json
    --variables capabilities/resolutions/1280x1024.json
```

Tox

With tox:

```
$ pip install tox
$ tox -epy36 -- -vvv --splinter-webdriver=remote \
    --variables=credentials/credentials_template.yml \
    --splinter-remote-url=http://USERNAME:ACCESS_KEY@hub.browserstack.com:80/wd/hub \
    --variables capabilities/os/WIN10.json
```

(continues on next page)

(continued from previous page)

```
    --variables capabilities/browsers/chrome/CHROME.json
    --variables capabilities/resolutions/1280x1024.json
```

1.1.3 Run tests with local browsers

You can launch tests based on local browsers instead of relying to a remote (SauceLabs, BrowserStack, TestingBot) or local grid (using Zalenium) using the `--splinter-webdriver firefox` option for example.

See https://github.com/pytest-dev/pytest-splinter#command-line-options

Supported browser options:

- firefox
- remote (you need to provide a value for the `--splinter-remote-url` option)
- chrome
- phantomjs

Using local browsers it's up to you the configuration of geckodriver, chromedriver, executable path settings, using the latest drivers (eg: https://github.com/mozilla/geckodriver/releases) and updated browser versions.

1.1.4 pytest-play ready!

`cookiecutter-qa` supports also pytest-play.

If you are not keen on programming or page objects you can run scenarios using a **json** format.

See `test_play.py` and `play.json`.

1.1.5 Credits

- heavily based on cookiecutter-pypackage: @audreyr's ultimate Python package project template.

1.1.6 Twitter

cookiecutter-qa tweets happens here:

- @davidemoro

1.1.7 Based on

COOKIECUTTER

1.1.8 Sponsored by

CHAPTER 2

Basics

2.1 Prompts

When you create a package, you are prompted to enter these values.

2.1.1 Templated Values

The following appear in various parts of your generated project.

full_name Your full name.

email Your email address.

github_username Your GitHub username.

project_name The name of your new Python package project. This is used in documentation, so spaces and any characters are fine here.

project_slug The namespace of your Python package. This should be Python import-friendly. Typically, it is the slugified version of project_name.

project_short_description A 1-sentence description of what your Python package does.

version The starting version number of the package.

create_author_file Creates an author file

open_source_license Select an open source license or not open source

command_line_interface Whether to create a console script using Click. Console script entry point will match the project_slug. Options: ['Click', "No command-line interface"]

base_url Your base url for your Splinter/Selenium tests

selenium_grid_url Your remote selenium grid url

testrail Upload test execution results to the Testrail (https://github.com/dubner/pytest-testrail) test management tool. If you don't have Testrail say `n` here

CHAPTER 3

Advanced Features

3.1 Console Script Setup

Optionally, your package can include a console script

3.1.1 How It Works

If the ‘command_line_interface’ option is set to [‘click’] during setup, cookiecutter will add a file ‘cli.py’ in the project_slug subdirectory. An entry point is added to setup.py that points to the main function in cli.py.

3.1.2 Usage

To use the console script in development:

```
pip install -e projectdir
```

‘projectdir’ should be the top level project directory with the setup.py file

The script will be generated with output for no arguments and –help.

--help	show help menu and exit

3.1.3 Known Issues

Installing the project in a development environment using:

```
python setup.py develop
```

will not set up the entry point correctly. This is a known issue with Click. The following will work as expected:

```
python setup.py install
pip install mypackage
```

With ‘mypackage’ adjusted to the specific project.

3.1.4 More Details

You can read more about Click at: http://click.pocoo.org/

CHAPTER 4

Indices and tables

- genindex
- modindex
- search