



tierra_qa Documentation

Release 0.1

Tierra QA team

Sep 29, 2017

Contents

1	Index	3
1.1	tierra_qa	3
1.2	Advanced	3
1.3	Contributors	6
1.4	Changelog	6
	Python Module Index	7

IMPORTANT Notes: tierra_qa will be no more maintained. Use **cookiecutter-qa** instead (<https://github.com/tierratelematics/cookiecutter-qa>)!

tierra_qa

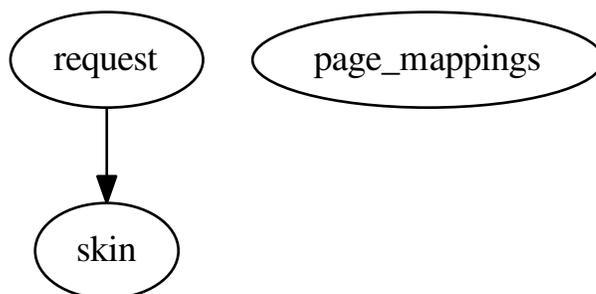
IMPORTANT Notes: tierra_qa will be no more maintained. Use **cookiecutter-qa** instead (<https://github.com/tierratelematics/cookiecutter-qa>)!

Advanced

Here you can see the technical documentation.

Fixture Diagrams

The following diagram shows the interactions between the `pytest` fixtures created in the `tierra_qa` package:



`tierra_qa.tests.conftest.pytestbdd_feature_base_dir()`
Feature files base directory.

`tierra_qa.tests.conftest.skin(request)`
This fixture provides the skin associated with the application on which starts the test session.

`tierra_qa.tests.conftest.default_pages()`
A mapping with the default page object class for each skin
It's up to you override this fixture with your settings.

For example:

```
DEFAULT_PAGES = {
    'skin1': 'mypackage.pages.BasePage',
}
```

`tierra_qa.tests.conftest.page_mappings()`
Returns the page mappings dictionary with all known page with:

- paths
- optional page object class (otherwise the default implementation will be used as fallback provided by `default_page_class()`)

See `tierra_qa.config` for further details.

Returns dictionary with all known pages

Return type dict

`tierra_qa.tests.conftest.splinter_screenshot_dir()`
Feature files base directory.

`tierra_qa.tests.conftest.splinter_driver_kwargs(splinter_webdriver)`
Webdriver kwargs.

Page mappings config

Here you can define all your page mappings with path and optionally a `class_page` if you want your specific page object implementation. By default the `tierra_qa.pages.BasePage` will be instantiated.

For example you could configure your own page object implementations:

```
PAGE_MAPPINGS = {
    'HomePage': {'path': '/'},
    'HelloPage': {'path': '/hello'},
    'AnotherPage': {'path': '/anotherpage', 'class_page': AnotherPage},
    'AnotherPage': {
        'path': '/myfleet/cnh/app/view/io_profile.html',
        'page_class': {'skin1': ListingPage,
                       'skin2': Listing2Page,
                       'fallback': ListingPage},
        'links': {'edit': 'DigitalInputEditPage'}},
}
```

This is only an example, you can also decide to add extra optional info such as `links` that tells you where the current page leads you to another page or whatever you want.

Base page implementation

This is where the base page implementation lives.

class `tierra_qa.pages.base.BasePage` (*driver, base_url=None, timeout=10, **url_kwargs*)

This is the base page to be used in `tierra_qa`.

Once you clone `tierra_qa` for your own qa project, you should:

- implement the `login`, `is_loggedin` and `logout` methods of this class
- add your own page object classes depending on your business logics inheriting from this base class

login (*username, password*)

This is the `login` method of the base page object.

It's up to you implement this method once you cloned `tierra_qa`.

Returns BasePage instance

Return type object

is_loggedin ()

This is the `is_loggedin` method of the base page object.

It's up to you implement this method once you cloned `tierra_qa`.

Returns True if you are logged in or False

Return type bool

username ()

This is the `username` method of the base page object.

It's up to you implement this method once you cloned `tierra_qa`.

Returns the username or None

Return type string or None

logout ()

This is the `logout` method of the base page object.

It's up to you implement this method once you cloned `tierra_qa`.

Returns BasePage instance

Return type object

BDD tests examples

Here you can see a BDD test example:

```
Feature: Login
  A user login try to login to the application

  @UI @login
  Scenario: Successful login
    Given I am logged in as Administrator
    Then I am logged in
```

where `Administrator` stands for the user id.

You have to provide the `Administrator` username and password providing them in the `credentials_template.yml`. For example:

```
skins:
  skin1:
    base_url: http://www.tierratelematics.com/
    credentials:
      Administrator:
        username: admin
        password: admin
```

Contributors

- [Full contributors list](#)

Changelog

0.0.1dev (unreleased)

- Initial release

t

tierra_qa.config, 4
tierra_qa.pages.base, 4
tierra_qa.tests.conftest, 3

B

BasePage (class in tierra_qa.pages.base), 5

D

default_pages() (in module tierra_qa.tests.conftest), 4

I

is_loggedin() (tierra_qa.pages.base.BasePage method), 5

L

login() (tierra_qa.pages.base.BasePage method), 5

logout() (tierra_qa.pages.base.BasePage method), 5

P

page_mappings() (in module tierra_qa.tests.conftest), 4

pytestbdd_feature_base_dir() (in module tierra_qa.tests.conftest), 3

S

skin() (in module tierra_qa.tests.conftest), 4

splinter_driver_kwargs() (in module tierra_qa.tests.conftest), 4

splinter_screenshot_dir() (in module tierra_qa.tests.conftest), 4

T

tierra_qa.config (module), 4

tierra_qa.pages.base (module), 4

tierra_qa.tests.conftest (module), 3

U

username() (tierra_qa.pages.base.BasePage method), 5