

---

# **kwalitee Documentation**

***Release 0.2.0***

**CERN**

February 25, 2015



<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Installation</b>	<b>5</b>
2.1	Versions . . . . .	5
2.2	Installation of the command-line interface . . . . .	5
2.3	Deployment of the web server . . . . .	6
<b>3</b>	<b>Command-line interface</b>	<b>9</b>
3.1	Help . . . . .	9
3.2	check . . . . .	9
3.3	githooks . . . . .	9
3.4	account . . . . .	10
3.5	repository . . . . .	11
<b>4</b>	<b>Testing</b>	<b>13</b>
4.1	Writing tests . . . . .	13
4.2	Fixtures . . . . .	13
4.3	Other tools . . . . .	13
<b>5</b>	<b>Changelog</b>	<b>15</b>
5.1	Release 0.2.0: <i>The next big thing</i> ® . . . . .	15
5.2	Release 0.1.0: <i>The playground</i> . . . . .	15
<b>6</b>	<b>API</b>	<b>17</b>
6.1	kwalitee package . . . . .	17
<b>7</b>	<b>Authors</b>	<b>29</b>
7.1	Contributors . . . . .	29
<b>8</b>	<b>Indices and tables</b>	<b>31</b>
	<b>Python Module Index</b>	<b>33</b>



Contents:



---

# Introduction

---

Kwalitee is a tool that runs static analysis checks on invenio and invenio-related repositories. It can be used as a web service using the Github API or as a git hook from the command line.

It aims at slowly, but steadily enforce good practices regarding commit message formatting, code layout (PEP8), documentation (PEP257) and help the integrators doing their job without having to worry about recurrent mistakes.

It relies on and thanks the following softwares and libraries:

- [pyflakes](#),
- [PEP8](#),
- [PEP257](#),
- [RQ](#),
- [Flask](#),
- [Requests](#) and
- [SQLAlchemy](#).





---

## Installation

---

### 2.1 Versions

We recommend you to use the stable version unless you want to contribute.

#### 2.1.1 Stable version

kwalitee is on PyPI so all you need is:

```
$ pip install --user kwalitee
```

#### 2.1.2 Development version

```
$ git clone https://github.com/jirikuncar/kwalitee
$ cd kwalitee
$ pip install --user -r requirements.txt
```

### 2.2 Installation of the command-line interface

Kwalitee can be used as a *Command-line interface*, which has some handy features like the *githooks* and the *Messages* checks.

By default the messages checks will try to use GitPython but we are recommending you to install and use `pygit2`. As, it has not stable version yet, the installation requires some work.

#### 2.2.1 Ubuntu

```
$ sudo apt-add-repository ppa:dennis/python
$ sudo apt-get update
$ sudo apt-get install python-dev libffi-dev libgit2
$ pip install cffi pygit2
```

If you don't find a suitable version using `ppa:dennis/python`, you can always install it manually via `cmake`.

## 2.2.2 OSX

The important detail here is to use the same version for `libgit2` **and** `pygit2`. [Homebrew](#) is a way to get it working.

```
$ brew update
$ brew install libgit2 # currently 0.21.0 (2014-07-28)
$ pip install pygit2
```

## 2.3 Deployment of the web server

kwalitee is composed of a WSGI server and a worker to handle the long tasks asynchronously.

### 2.3.1 Requirements

- A WSGI web server, we recommend [nginx](#) + [uWSGI](#) but anything may do. Apache + [mod\\_wsgi](#) is working as well.
- A Redis server.

### 2.3.2 Configuration

There are not many things you have to configure on a fresh installation, here are the few you should be aware of.

#### **ACCESS\_TOKEN**

Is the API token for your user on Github. You need it in order to be able to publish comments and statuses.

It can be defined per account as well. See: [Command-line interface](#) and `kwalitee.models.Account`.

#### **AUTO\_CREATE**

Unless `AUTO_CREATE` is set to true, you'll have to enable the repositories individually to authorize events from Github.

```
$ kwalitee repository add invenio/test
invenio/test is now allowed to webhook kwalitee!
```

### 2.3.3 WSGI application

The web application can be served using [nginx](#) + [uWSGI](#) or [gunicorn](#).

#### **Development server**

Using Flask Script's [Server](#), you can run it without any external servers or libraries.

```
$ kwalitee runserver
```

## uWSGI

This configuration file will serve the applicatino on port 8000.

```
; uwsgi.ini
[uwsgi]

http = 0.0.0.0:8000
master = true

processes = 4
die-on-term = true
vacuum = true
max-requests = 100

chdir = <VIRTUALENV>/opt/kwalitee
virtualenv = <VIRTUALENV>
module = kwalitee.wsgi:application
touch-reload = uwsgi.ini

enable-threads = true
```

And start it this way.

```
$ uwsgi --init uwsgi.ini
```

See more on [uWSGI](#) documentation.

## nginx + uWSGI

### TODO

### 2.3.4 Worker

A simple way to run the worker is the following. It works well for development and/or debug purposes. Consider relying on a daemon supervisor like: [upstart](#), [systemd](#), [runit](#) or [supervisord](#).

```
$ python -m kwalitee.worker
```

### Upstart (Ubuntu)

The worker can also be handled using [upstart](#). Here is the configuration for it. [VirtualEnv](#) is a clean way to set everything up and is recommended.

```
# /etc/init/<myservice>.conf
description "Kwalitee RQ worker"

respawn
respawn limit 15 5
console log
setuid <USER>
setgid <GROUP>

exec /usr/bin/python -m kwalitee.worker
# Or if you've set it up in a virtualenv
#exec <VIRTUALENV>/bin/python -m kwalitee.worker
```

Then, you can manage it using upstart like anything else.

```
$ sudo start <myservice>  
$ sudo stop <myservice>
```

---

## Command-line interface

---

*kwalitee* comes with a command-line tool that goes by the name of *kwalitee*. If you've installed it using the `--user` option, you'll have to add `~/.local/bin` to your path. Otherwise, you should be able to call it without any trouble.

```
$ export PATH+=:~/.local/bin
```

### 3.1 Help

The command line tool is able to give you direct little help.

```
$ kwalitee --help
```

### 3.2 check

Utility to run various *kwalitee* checks in your repository.

#### 3.2.1 Messages

#### 3.2.2 message

Runs the checks on the existing commits.

```
$ kwalitee check message master..
```

### 3.3 githooks

This tool can install or uninstall some hooks into the *current* git repository.

**See also:**

- `kwalitee.cli.githooks`
- `kwalitee.hooks`

### 3.3.1 Hooks

#### `pre-commit`

Runs the checks on the files about to be committed.

#### `prepare-commit-msg`

Based on the state of the commit create a commit message to be filled in.

**See also:**

`kwalitee.config.COMMIT_MSG_TEMPLATE`

#### `post-commit`

Verifies that the commit message passes the checks.

### 3.3.2 Installation

```
$ kwalitee install
```

### 3.3.3 Uninstallation

```
$ kwalitee uninstall
```

## 3.4 account

Utility to manage to user account that have registered.

### 3.4.1 Listing

```
$ kwalitee account list
```

### 3.4.2 Creation and modification

Creation and modification are using the `add` command. You can alter the user's email and its GitHub API token. Any user with a token will have the comments posted on his repository made using the token's account instead of the default one.

```
$ kwalitee account add <ACCOUNT>
```

```
$ kwalitee account add <ACCOUNT> --email <EMAIL> --token <TOKEN>
```

**See also:**

`kwalitee.config.ACCESS_TOKEN`

### 3.4.3 Deletion

Deletion is permanent and it deletes everything belonging to the given account.

```
$ kwalitee account remove <ACCOUNT>
```

## 3.5 repository

Utility to manage to user's repositories.

### 3.5.1 Listing

```
$ kwalitee repository list
```

### 3.5.2 Creation

```
$ kwalitee repository add <ACCOUNT>/<REPOSITORY>
```

### 3.5.3 Deletion

Deletion is permanent and it deletes everything belonging to the given repository.

```
$ kwalitee repository remove <ACCOUNT>/<REPOSITORY>
```





---

## Testing

---

Running the tests are as simple as:

```
$ python setup.py test
```

The code coverage can be output by passing some arguments to `py.test`.

```
$ python setup.py test -a "tests --cov kwalitee --cov-config .coveragerc"
# html report
$ python setup.py test -a "tests --cov kwalitee --cov-report html"
```

Ditto for running only one test or debugging with `pdb`.

```
$ python setup.py test -a tests/tests_ping.py
$ python setup.py test -a tests/tests_ping.py::test_ping
$ python setup.py test -a "tests --pdb"
```

### 4.1 Writing tests

The tests are using `PyHamcrest` for its richness and the nice default output provided. To be consistent, avoid using `unittest` or bare `assert`.

### 4.2 Fixtures

Fixtures are provided by the very powerful `py.test`. Take a look at the fixtures defined in the *confest.py* files.

### 4.3 Other tools

#### 4.3.1 HTTPretty

`HTTPretty` is a HTTP client mock library which lets you define your own custom answers and status code as well as testing which calls have been made.

### 4.3.2 mock

`mock` is used to mock file opening (`open()`) and inspect the content that was written in it without having to create temporary files with `tempfile`.

---

## Changelog

---

This file describes the set of features of each version as well as the incompatibilities.

### 5.1 Release 0.2.0: *The next big thing®*

This version uses a database (SQLite) for persistence.

- Support for Docker. (Jiří)
- Support for `.kwalitee.yml` configuration per repository. (Haris)
- Cli for checking changed files `kwalitee check files`. (Jiří)
- Cli for checking commit messages `kwalitee check message`. (Jiří)
- Support of `push` events. (Yoan)
- Support for multiple repositories. (Yoan)
- Support for multiple users. (Yoan)
- Alembic setup for upcoming migrations (Yoan)
- New Sphinx documentation. (Yoan)
- Fixes double commenting bug. (Yoan)

#### 5.1.1 Incompatibilities

- The commit statuses are still accessible but are not migrated to the database.
- Previously created git hooks will have to be uninstalled and re-installed as the Flask application is not always created.

### 5.2 Release 0.1.0: *The playground*

Initial version. It supports `pull request` events on one repository and will perform checks on the commit message and files.

- Commit message checks. (Jiří)
- Git hooks. (Lars)

- PEP8 checks. (Yoan)
- PEP257 checks. (Yoan)
- PyFlakes checks. (Yoan)
- License checks. (Yoan)
- Asynchronous checks using RQ. (Yoan)
- New unit tests. (Yoan)
- Auto labelling of the pull requests. (Yoan)
- Skip work in progress (wip) pull requests. (Yoan)

## 6.1 kwalitee package

### 6.1.1 Subpackages

#### kwalitee.cli package

##### Submodules

##### kwalitee.cli.account module

Command-line tools for managing accounts.

`kwalitee.cli.account.add(account, email=None, token=None)`  
Add/modify an account.

`kwalitee.cli.account.list()`  
List all the repositories.

`kwalitee.cli.account.remove(account)`  
Remove an account (and its repositories).

##### kwalitee.cli.githooks module

Command-line tools for the git hooks.

`kwalitee.cli.githooks.install(force=False)`  
Install git hooks.

`kwalitee.cli.githooks.uninstall()`  
Uninstall git hooks.

##### kwalitee.cli.repository module

Command-line tools for managing repositories.

`kwalitee.cli.repository.add(repository)`  
Add a repository to the list of authorized ones.

```
kwalitee.cli.repository.list()
```

List all the repositories.

```
kwalitee.cli.repository.remove(repository)
```

Remove a repository from the list of authorized ones.

## Module contents

Command line interfaces entrypoints.

```
kwalitee.cli.main()
```

Running the manager.

## 6.1.2 Submodules

### 6.1.3 kwalitee.config module

kwalitee base configuration.

To change it, put a config.py into your instance.

```
kwalitee.config.CONTEXT
```

Context for Github.

**See also:**

[Github API Statuses](#)

```
kwalitee.config.COMPONENTS
```

List of supported components.

**See also:**

```
kwalitee.check_message()
```

```
kwalitee.config.ACCESS_TOKEN
```

Github access token. Used to post statuses and comments. It MUST be set.

```
kwalitee.config.AUTO_CREATE
```

Allow anyone to add its repository by setting up the webhook on Github.

**Default:** False

```
kwalitee.config.CHECK_COMMIT_MESSAGES
```

Enable the `commit message checks`.

**Default:** True

```
kwalitee.config.CHECK_WIP
```

Enable the *work-in-progress* pull requests checks. Disabled by default.

**Default:** False

```
kwalitee.config.CHECK_LICENSE
```

Enable the `license checks`.

**Default:** True

```
kwalitee.config.CHECK_PEP8
```

Enable the `PEP8 checks`.

**Default:** True

`kwalitee.config.CHECK_PEP257`

Enable the `PEP257` checks.

**Default:** `True`

`kwalitee.config.CHECK_PYFLAKES`

Enable the PyFlakes checks. PEP8 checks are required.

**Default:** `True`

`kwalitee.config.IGNORE`

Error codes to ignore.

**Default:** `['E123', 'E226', 'E24', 'E501', 'E265']`

`kwalitee.config.SELECT`

Error codes to specially enable.

**Default:** `[]`

`kwalitee.config.WORKER_TIMEOUT`

Background worker job time window.

Any job taking longer than that will be killed.

RQ default timeout is 180 seconds

`kwalitee.config.MIN_REVIEWERS`

Minimum number of reviewers for `py:func:message check <.kwalitee.check_message>`.

**Default:** `3`

`kwalitee.config.LABEL_WIP`

Label to apply for a *work-in-progress* pull request.

**Default:** `"in_work"`

`kwalitee.config.LABEL_REVIEW`

Label to apply for a pull request that needs more reviewers.

**Default:** `"in_review"`

`kwalitee.config.LABEL_READY`

Label to apply for a pull request that passes all the checks.

**Default:** `"in_integration"`

`kwalitee.config.ALT_SIGNATURES = ('Reported-by',)`

Alternative signatures recognized but not counted as reviewers.

`kwalitee.config.COMMIT_MSG_TEMPLATE = '{component}: description (max 50 chars, using nouns)\n\n* Detailed description'`

Template used to generate the commit message from the git hook.

`kwalitee.config.GITHUB = 'https://github.com/'`

Github base URL.

`kwalitee.config.GITHUB_REPO = 'https://github.com/{account}/{repository}'`

Github repository URL template.

`kwalitee.config.HOOK_TEMPLATE = '#!/usr/bin/env python\n# -*- coding: utf-8 -*-\nimport sys\nfrom kwalitee import cr'`

Template used to generate the git hooks, customize at will.

`kwalitee.config.PEP257_MATCH = '(?!test_).*\\.py'`

Files checked for PEP257 conformance.

`kwalitee.config.PEP257_MATCH_DIR = '[^\\].*'`

Directories checks for PEP257 conformance.

`kwalitee.config.SIGNATURES = ('Signed-off-by', 'Co-authored-by', 'Tested-by', 'Reviewed-by', 'Acked-by')`

Authors and reviewers signatures.

`kwalitee.config.TRUSTED_DEVELOPERS = []`

Super developers who's code never fail.

## 6.1.4 kwalitee.factory module

kwalitee factory for Flask application.

```
class kwalitee.factory.ShaConverter(map)
    Bases: werkzeug.routing.BaseConverter

    Werkzeug routing converter for sha-1 (truncated or full).

    regex = u'(!/)(?:[a-fA-F0-9]{40}|[a-fA-F0-9]{7})'
    weight = 150
```

`kwalitee.factory.create_app(name=None, config=None)`  
Create the Flask application.

## 6.1.5 kwalitee.hooks module

Git hooks.

`kwalitee.hooks.commit_msg_hook(argv)`  
Hook: for checking commit message (prevent commit).

`kwalitee.hooks.post_commit_hook(argv=None)`  
Hook: for checking commit message.

`kwalitee.hooks.pre_commit_hook(argv=None)`  
Hook: checking the staged files.

`kwalitee.hooks.prepare_commit_msg_hook(argv)`  
Hook: prepare a commit message.

`kwalitee.hooks.run(command, raw_output=False)`  
Run a command using subprocess.

### Parameters

- **command** (*str*) – command line to be run
- **raw\_output** (*bool*) – does not attempt to convert the output as unicode

**Returns** error code, output (stdout) and error (stderr)

**Return type** tuple

## 6.1.6 kwalitee.kwalitee module

Kwalitee checks for PEP8, PEP257, PyFlakes and License.

`kwalitee.kwalitee.SUPPORTED_FILES = (u'.py', u'.html', u'.tpl', u'.js', u'.jsx', u'.css', u'.less')`  
Supported file types.



`kwalitee.kwalitee.check_file(filename, **kwargs)`

Perform static analysis on the given file.

**See also:**

- `SUPPORTED_FILES`
- `check_pep8()`
- `check_pep257()`
- and `check_license()`

**Parameters** `filename` (*str*) – path of file to check.

**Returns** errors sorted by line number

**Return type** list

`kwalitee.kwalitee.check_license(filename, **kwargs)`

Perform a license check on the given file.

The license format should be commented using # and live at the top of the file. Also, the year should be the current one.

**Parameters**

- **filename** (*str*) – path of file to check.
- **year** (*int*) – default current year
- **ignore** (*list*) – codes to ignore, e.g. ('L100', 'L101')
- **python\_style** (*bool*) – False for JavaScript or CSS files

**Returns** errors

**Return type** list

`kwalitee.kwalitee.check_message(message, **kwargs)`

Check the message format.

**Rules:**

- the first line must start by a component name
- and a short description (52 chars),
- then bullet points are expected
- and finally signatures.

**Parameters**

- **components** (*list*) – components, e.g. ('auth', 'utils', 'misc')
- **signatures** (*list*) – signatures, e.g. ('Signed-off-by', 'Reviewed-by')
- **alt\_signatures** (*list*) – alternative signatures, e.g. ('Tested-by',)
- **trusted** (*list*) – optional list of reviewers, e.g. ('john.doe@foo.org',)
- **max\_length** (*int*) – optional maximum line length (by default: 72)
- **max\_first\_line** (*int*) – optional maximum first line length (by default: 50)
- **allow\_empty** (*bool*) – optional way to allow empty message (by default: False)

**Returns** errors sorted by line number

**Return type** list

`kwalitee.kwalitee.check_pep257(filename, **kwargs)`

Perform static analysis on the given file docstrings.

**Parameters**

- **filename** (*str*) – path of file to check.
- **ignore** (*list*) – codes to ignore, e.g. ('D400',)
- **match** (*str*) – regex the filename has to match to be checked
- **match\_dir** (*str*) – regex everydir in path should match to be checked

**Returns** errors

**Return type** list

**See also:**

[GreenSteam/pep257](#)

`kwalitee.kwalitee.check_pep8(filename, **kwargs)`

Perform static analysis on the given file.

**Parameters**

- **filename** (*str*) – path of file to check.
- **ignore** (*list*) – codes to ignore, e.g. ('E111', 'E123')
- **select** (*list*) – codes to explicitly select.
- **pyflakes** (*bool*) – run the pyflakes checks too (default True)

**Returns** errors

**Return type** list

**See also:**

[pep8.Checker](#)

`kwalitee.kwalitee.check_whitespaces(self, definition, docstring)`

D290: White spaces around doctring should be trimmed.

`kwalitee.kwalitee.get_options(config)`

Build the options from the Flask config.

## 6.1.7 kwalitee.models module

Database models to persist the data over time.

**class** `kwalitee.models.Account` (*name, email=None, token=None*)

Bases: `flask_sqlalchemy.Model`

Github account.

**classmethod** `create` (*name, email=None, token=None*)

Create and commit new account.

**created\_at**

Date of creation.

```
email
    Email.

classmethod find_or_create (name, email=None, token=None)
    Find or create an account.

id
    Identifier

name
    Username.

token
    API Token.

classmethod update_or_create (name, email=None, token=None)
    Modify or create an account.

updated_at
    Date of last modification.

class kwalitee.models.BranchStatus (commit, name, url, content=None)
    Bases: flask_sqlalchemy.Model

    Status of a pull request.

commit

commit_id

content
    Get the content of the status.

created_at

errors
    Get the number of errors found.

classmethod find_or_create (commit, name, url, content=None)
    Find or create a commit status.

get_content ()
    Get the content of the status.

id

is_pending ()
    Return True is the commit status hasn't been checked yet.

name

set_content (value)
    Set the content of the status.

state
    Get the state.

updated_at

url

class kwalitee.models.CommitStatus (repository, sha, url, content=None)
    Bases: flask_sqlalchemy.Model

    Status of a push.
```

```

content
    Get the content of the status.

created_at

errors
    Get the number of errors found.

classmethod find_or_create (repository, sha, url)
    Find or create a commit status.

get_content ()
    Get the content of the status.

id

is_pending ()
    Return True is the commit status hasn't been checked yet.

repository

repository_id

set_content (value)
    Set the content of the status.

sha

state
    Get the state.

updated_at

url

class kwalitee.models.Repository (owner, name)
    Bases: flask_sqlalchemy.Model

    Github repository.

    created_at

    classmethod find_or_create (owner, name)
        Find or create a repository.

    fullname
        Get the fullname of the repository.

    id

    name

    owner

    owner_id

    updated_at

kwalitee.models.init_app (app)
    Initialize the Flask app with db.

```

### 6.1.8 kwalitee.tasks module

Tasks to run on the worker.

`kwalitee.tasks.get_headers(repository, config)`

Get the HTTP headers for the GitHub api.

This is required to post comments on GitHub on your behalf. Please update your configuration accordingly.

```
ACCESS_TOKEN = "deadbeef..."
```

It can also be overwritten per user.

```
$ kwalitee account add username --token=deadbeef...
```

**Returns** HTTP headers

**Return type** dict

`kwalitee.tasks.pull_request(branch_status_id, pull_request_url, status_url, config)`

Performing all the tests on the pull request.

Then pings back the given status\_url and update the issue labels.

**Parameters**

- **branch\_status\_id** (*int*) – identifier of the branch status.
- **pull\_request\_url** (*str*) – github api pull request
- **status\_url** (*str*) – github api status url
- **config** (*dict*) – configuration dictionary

**Returns** status body and applied labels

**Return type** dict

`kwalitee.tasks.push(commit_status_id, commit_url, status_url, config)`

Performing all the tests on the commit.

**Parameters**

- **commit\_status\_id** – identifier of the branch status
- **commit\_url** – github api commit url
- **status\_url** – github api status url
- **config** – configuration dictionary

## 6.1.9 kwalitee.views module

Views like in MTV.

`kwalitee.views.account(account)`

Display the repositories linked with one account.

**Parameters** **account** – name of the account

`kwalitee.views.branch(account, repository, branch)`

Display the statuses of a branch.

**Parameters**

- **account** – name of the owner
- **repository** – name of the repository

- **branch** – name of the branch

`kwalitee.views.branch_status(account, repository, branch, sha)`

Display the status of a pull request.

**Parameters**

- **account** – name of the owner
- **repository** – name of the repository
- **branch** – name of the branch
- **sha** – commit identifier of the commit related with the branch

`kwalitee.views.commit(account, repository, sha)`

Display the status of a commit.

**Parameters**

- **account** – name of the owner
- **repository** – name of the repository
- **sha** – identifier of the commit

`kwalitee.views.index()`

Homepage that lists the accounts.

`kwalitee.views.payload()`

Handle the GitHub events.

**See also:**

*Event Types* <<https://developer.github.com/v3/activity/events/types/>>

`kwalitee.views.repository(account, repository, limit=50)`

Display the recents commits and branches of a repository.

**Parameters**

- **account** – name of the owner
- **repository** – name of the repository
- **limit** – size of the commit window

`kwalitee.views.status(sha)`

Show the status of a commit.

**deprecated** static files aren't used anymore. To be removed at some point.

**Parameters** **sha** – identifier of a commit.

## 6.1.10 kwalitee.worker module

Initialize Redis and setups the RQ worker.

`kwalitee.worker.init_app(app)`

Initialize the RQ queue.

## 6.1.11 kwalitee.wsgi module

WSGI application with debug middleware if in debug mode.

### 6.1.12 Module contents

kwalitee Flask application and git hooks.

`kwalitee.create_app` (*name=None, config=None*)

Create the Flask application.





---

### Authors

---

kwality is developed to support developers of [Invenio](#) digital library software.

Contact us at [info@invenio-software.org](mailto:info@invenio-software.org).

### 7.1 Contributors

- Jiri Kuncar <[jiri.kuncar@cern.ch](mailto:jiri.kuncar@cern.ch)>
- Yoan Blanc <[yoan@dosimple.ch](mailto:yoan@dosimple.ch)>
- Lars Holm Nielsen <[lars.holm.nielsen@cern.ch](mailto:lars.holm.nielsen@cern.ch)>
- Mateusz Susik <[mateuszsusik@gmail.com](mailto:mateuszsusik@gmail.com)>



---

## Indices and tables

---

- *genindex*
- *modindex*
- *search*



## k

- `kwalitee`, [27](#)
- `kwalitee.cli`, [18](#)
- `kwalitee.cli.account`, [17](#)
- `kwalitee.cli.githooks`, [17](#)
- `kwalitee.cli.repository`, [17](#)
- `kwalitee.config`, [18](#)
- `kwalitee.factory`, [20](#)
- `kwalitee.hooks`, [20](#)
- `kwalitee.kwalitee`, [20](#)
- `kwalitee.models`, [22](#)
- `kwalitee.tasks`, [24](#)
- `kwalitee.views`, [25](#)
- `kwalitee.worker`, [26](#)
- `kwalitee.wsgi`, [26](#)



## A

ACCESS\_TOKEN (in module kwalitee.config), 18  
Account (class in kwalitee.models), 22  
account() (in module kwalitee.views), 25  
add() (in module kwalitee.cli.account), 17  
add() (in module kwalitee.cli.repository), 17  
ALT\_SIGNATURES (in module kwalitee.config), 19  
AUTO\_CREATE (in module kwalitee.config), 18

## B

branch() (in module kwalitee.views), 25  
branch\_status() (in module kwalitee.views), 26  
BranchStatus (class in kwalitee.models), 23

## C

CHECK\_COMMIT\_MESSAGES (in module kwalitee.config), 18  
check\_file() (in module kwalitee.kwalitee), 20  
CHECK\_LICENSE (in module kwalitee.config), 18  
check\_license() (in module kwalitee.kwalitee), 21  
check\_message() (in module kwalitee.kwalitee), 21  
CHECK\_PEP257 (in module kwalitee.config), 18  
check\_pep257() (in module kwalitee.kwalitee), 22  
CHECK\_PEP8 (in module kwalitee.config), 18  
check\_pep8() (in module kwalitee.kwalitee), 22  
CHECK\_PYFLAKES (in module kwalitee.config), 19  
check\_whitespaces() (in module kwalitee.kwalitee), 22  
CHECK\_WIP (in module kwalitee.config), 18  
commit (kwalitee.models.BranchStatus attribute), 23  
commit() (in module kwalitee.views), 26  
commit\_id (kwalitee.models.BranchStatus attribute), 23  
commit\_msg\_hook() (in module kwalitee.hooks), 20  
COMMIT\_MSG\_TEMPLATE (in module kwalitee.config), 19  
CommitStatus (class in kwalitee.models), 23  
COMPONENTS (in module kwalitee.config), 18  
content (kwalitee.models.BranchStatus attribute), 23  
content (kwalitee.models.CommitStatus attribute), 23  
CONTEXT (in module kwalitee.config), 18  
create() (kwalitee.models.Account class method), 22

create\_app() (in module kwalitee), 27  
create\_app() (in module kwalitee.factory), 20  
created\_at (kwalitee.models.Account attribute), 22  
created\_at (kwalitee.models.BranchStatus attribute), 23  
created\_at (kwalitee.models.CommitStatus attribute), 24  
created\_at (kwalitee.models.Repository attribute), 24

## E

email (kwalitee.models.Account attribute), 22  
errors (kwalitee.models.BranchStatus attribute), 23  
errors (kwalitee.models.CommitStatus attribute), 24

## F

find\_or\_create() (kwalitee.models.Account class method), 23  
find\_or\_create() (kwalitee.models.BranchStatus class method), 23  
find\_or\_create() (kwalitee.models.CommitStatus class method), 24  
find\_or\_create() (kwalitee.models.Repository class method), 24  
fullname (kwalitee.models.Repository attribute), 24

## G

get\_content() (kwalitee.models.BranchStatus method), 23  
get\_content() (kwalitee.models.CommitStatus method), 24  
get\_headers() (in module kwalitee.tasks), 24  
get\_options() (in module kwalitee.kwalitee), 22  
GITHUB (in module kwalitee.config), 19  
GITHUB\_REPO (in module kwalitee.config), 19

## H

HOOK\_TEMPLATE (in module kwalitee.config), 19

## I

id (kwalitee.models.Account attribute), 23  
id (kwalitee.models.BranchStatus attribute), 23  
id (kwalitee.models.CommitStatus attribute), 24  
id (kwalitee.models.Repository attribute), 24

IGNORE (in module kwalitee.config), 19  
 index() (in module kwalitee.views), 26  
 init\_app() (in module kwalitee.models), 24  
 init\_app() (in module kwalitee.worker), 26  
 install() (in module kwalitee.cli.githooks), 17  
 is\_pending() (kwalitee.models.BranchStatus method), 23  
 is\_pending() (kwalitee.models.CommitStatus method), 24

## K

kwalitee (module), 27  
 kwalitee.cli (module), 18  
 kwalitee.cli.account (module), 17  
 kwalitee.cli.githooks (module), 17  
 kwalitee.cli.repository (module), 17  
 kwalitee.config (module), 18  
 kwalitee.factory (module), 20  
 kwalitee.hooks (module), 20  
 kwalitee.kwalitee (module), 20  
 kwalitee.models (module), 22  
 kwalitee.tasks (module), 24  
 kwalitee.views (module), 25  
 kwalitee.worker (module), 26  
 kwalitee.wsgi (module), 26

## L

LABEL\_READY (in module kwalitee.config), 19  
 LABEL\_REVIEW (in module kwalitee.config), 19  
 LABEL\_WIP (in module kwalitee.config), 19  
 list() (in module kwalitee.cli.account), 17  
 list() (in module kwalitee.cli.repository), 17

## M

main() (in module kwalitee.cli), 18  
 MIN\_REVIEWERS (in module kwalitee.config), 19

## N

name (kwalitee.models.Account attribute), 23  
 name (kwalitee.models.BranchStatus attribute), 23  
 name (kwalitee.models.Repository attribute), 24

## O

owner (kwalitee.models.Repository attribute), 24  
 owner\_id (kwalitee.models.Repository attribute), 24

## P

payload() (in module kwalitee.views), 26  
 PEP257\_MATCH (in module kwalitee.config), 19  
 PEP257\_MATCH\_DIR (in module kwalitee.config), 19  
 post\_commit\_hook() (in module kwalitee.hooks), 20  
 pre\_commit\_hook() (in module kwalitee.hooks), 20  
 prepare\_commit\_msg\_hook() (in module kwalitee.hooks), 20

pull\_request() (in module kwalitee.tasks), 25  
 push() (in module kwalitee.tasks), 25

## R

regex (kwalitee.factory.ShaConverter attribute), 20  
 remove() (in module kwalitee.cli.account), 17  
 remove() (in module kwalitee.cli.repository), 18  
 Repository (class in kwalitee.models), 24  
 repository (kwalitee.models.CommitStatus attribute), 24  
 repository() (in module kwalitee.views), 26  
 repository\_id (kwalitee.models.CommitStatus attribute), 24  
 run() (in module kwalitee.hooks), 20

## S

SELECT (in module kwalitee.config), 19  
 set\_content() (kwalitee.models.BranchStatus method), 23  
 set\_content() (kwalitee.models.CommitStatus method), 24  
 sha (kwalitee.models.CommitStatus attribute), 24  
 ShaConverter (class in kwalitee.factory), 20  
 SIGNATURES (in module kwalitee.config), 20  
 state (kwalitee.models.BranchStatus attribute), 23  
 state (kwalitee.models.CommitStatus attribute), 24  
 status() (in module kwalitee.views), 26  
 SUPPORTED\_FILES (in module kwalitee.kwalitee), 20

## T

token (kwalitee.models.Account attribute), 23  
 TRUSTED\_DEVELOPERS (in module kwalitee.config), 20

## U

uninstall() (in module kwalitee.cli.githooks), 17  
 update\_or\_create() (kwalitee.models.Account class method), 23  
 updated\_at (kwalitee.models.Account attribute), 23  
 updated\_at (kwalitee.models.BranchStatus attribute), 23  
 updated\_at (kwalitee.models.CommitStatus attribute), 24  
 updated\_at (kwalitee.models.Repository attribute), 24  
 url (kwalitee.models.BranchStatus attribute), 23  
 url (kwalitee.models.CommitStatus attribute), 24

## W

weight (kwalitee.factory.ShaConverter attribute), 20  
 WORKER\_TIMEOUT (in module kwalitee.config), 19