
turses Documentation

Release 0.3.1

Alejandro Gómez, Nicolas Paris

Apr 25, 2017

Contents

1	Features	3
2	User Guide	5
2.1	Installation	5
2.2	Configuration	6
3	Dev Guide	15
3.1	Set up the development environment	15
3.2	Internals	16
3.3	Authors	19
3.4	Changelog	20
4	Screenshots	29
	Python Module Index	31

A Twitter client for the console.

CHAPTER 1

Features

- Multiple timelines (buffers)
- Multi-column
- Tweet, Reply, Retweet, Delete tweet
- Follow/Unfollow
- Favorite/Unfavorite
- Direct Messages
- Open URLs in browser
- Thread view
- Unread count
- Search
- View user's tweets
- Fully customizable
- Multiple accounts
- View user profile

This part of the documentation will help you using `turses`.

Installation

From source

`turses` is evolving fast, you can find the [latest version on GitHub](#).

You can either clone the public repository:

```
$ git clone git://github.com/alejandrogomez/turses.git
```

Download the [tarball](#):

```
$ curl -OL https://github.com/alejandrogomez/turses/tarball/master
```

Or, download the [zipball](#):

```
$ curl -OL https://github.com/alejandrogomez/turses/zipball/master
```

Once you have a copy of the source, you can embed it in your Python package, or install it into your site-packages easily:

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

Distribute & pip

Installing is simple with [pip](#):

```
$ pip install turses
```

or (but you should consider using pip):

```
$ easy_install turses
```

Configuration

The configuration files are located on `$HOME/.turses` directory.

There is one mayor configuration file in turses:

config contains user preferences: colors, bindings, etc.

An one default token file:

token contains authentication token for the default user account

Each user account that is no the default one needs to be aliased and has its own token file `alias.token`.

To create an aliased account:

```
$ turses -a work
```

And, after authorizing `turses` to use that account, a token file named `work.token` will be created. Optionally you can create a `work.config` file for a configuration specific to that account.

Now, when you execute again:

```
$ turses -a work
```

you will be logged in with the previously stored credentials.

Here is an example with two accounts apart from the default one, aliased to `alice` and `bob`.

```
~
|+.turses/
| |-config
| |-alice.config
| |-token
| |-alice.token
| `--bob.token
|+...
|+...
|+...
```

If you want to generate a configuration file, you can do so executing:

```
$ turses -g /path/to/file
```

The configuration file is divided into sections, each of which is described below these lines.

Warning: The `timelines` section of the configuration has been deprecated, use `sessions` instead.

Sessions

The file with the session declarations is located on `$HOME/.turses/sessions`.

`sessions` is a ini-style configuration file in which each section represents the layout of a session. The `defaults` session is loaded when no other section is present.

Each section has only two options:

visible contains the timelines that will be visible stacked in columns

buffers contains the timelines that won't be visible but will be loaded

Warning: The `visible` option must be present for any session but `buffers` is optional.

For each option, you will define the timelines as a comma-separated list of their names. Here is a list with the valid names:

- `home` for the home timeline
- `mentions` for the mentions timeline
- `favorites` for the favorites timeline
- `messages` for the direct message timeline
- `own_tweets` for the timeline with your tweets
- `search:<query>` for searching timelines
- `hashtag:<query>` for searching a hashtag
- `user:<screen_name>` for a user's timeline
- `retweets_of_me` for the timeline with your retweeted tweets

Declaring a custom session is as easy as defining a section on the `sessions` file. As an example, let's define a session called `interactions`, in which we would only like to view our mentions, messages and what people are saying about `turses`; and load the home timeline in background:

```
[interactions]
visible = mentions, messages, search:turses, hashtag:turses
buffer = home
```

If you would like to load a session when starting `turses`, you must provide the name of the session as a command-line argument. You can start the session named `interactions` by executing:

```
$ turses -s interactions
# or, alternatively
$ turses --session interactions
```

Twitter

This section allows you to configure the settings related to the Twitter API.

`turses` communicates with Twitter over HTTPS by default but you can switch it off if you set it to false:

```
[twitter]
use_https = false
```

However, we recommend you to use HTTPS, especially in open WiFi networks where anybody could be sniffing packages and reading your personal information and communications.

The other available option is `update_frequency` which controls how often (in seconds) the timelines should be automatically updated.

An example configuration that updates the timelines every minute:

```
[twitter]
update_frequency = 60
```

Bindings

Almost every action within `turses` is configurable. The defaults resemble some of the bindings from the `vi` editor. To see an up-to-date description of all the available actions open the help buffer pressing `?`.

An example configuration with the motion keys assigned to the arrow keys:

```
[bindings]
up = up
down = down
left = left
right = right
```

`turses` uses the representation of keystrokes provided by `urwid` to map the bindings to actions.

Colors

You can change the colors of different elements of the UI in `turses`. The legal values for colors are listed in the [urwid wiki](#).

An example configuration that sets a magenta background and white foreground in the editor:

```
[colors]
editor = white
editor_bg = dark magenta
```

Styles

This section allows you to configure the styles for some of the UI elements in `turses`. Below is a description of all the configuration options in the section.

Templates

The templates allow you to configure how certain text is rendered in `turses`. The following templates are available:

- `header_template`: The header of a tweet.
- `dm_template`: The header of a direct message.
- `tab_template`: The text in a tab.

These templates contain variables enclosed between braces that are replaced by their corresponding value. Let's look at the defaults to see all the available variables within the templates:

```
[styles]
reply_indicator =
retweet_indicator =
header_template = {username}{retweeted}{retweeter} - {time}{reply} {retweet_count}
```

- `username`: The author of the tweet.
- `retweeted`: The value of `retweet_indicator` if the status is a retweet.
- `retweeter`: The name of the retweeter (if any).
- `time`: Relative time of the tweet.
- `reply`: The value of `reply_indicator` if the status is a reply.
- `retweet_count`: The number of retweets.

Warning: The `reply_indicator` and `retweet_indicator` values will be surrounded with spaces.

```
[styles]
dm_template = {sender_screen_name} => {recipient_screen_name} - {time}
```

- `sender_screen_name`: The sender of the message.
- `recipient_screen_name`: The recipient of the message.
- `time`: Relative time of the message.

```
[styles]
tab_template = {unread} {timeline_name}
```

- `unread`: Unread tweet count.
- `timeline_name`: The name of the timeline.

Tweets

You can configure how tweets are rendered. By default the statuses are enclosed in a box, but you can use a divider instead.

Here's how the default configuration for status styles looks like:

```
[styles]
box_around_status = true
```

If `box_around_status` is set to `true`, the tweets will be rendered as follows:

```

dev social sys www tiled 0 Off 27% 0:3
44 tweets | 19 mentions
NathanBowers & codinghorror - 4 minutes ago 2
  "What happens when you let greed and lawyers run amok over common sense and generosity." ht
alex_gaynor - 9 minutes ago
  Fuck yeah! http://t.co/kLlSaS0v
grcpriest - 13 minutes ago
  Está curioso, ha
eduo - 15 minutes ago
  Jajaja! Que bueno. FuckItJS es un gestor de codigo javascript que borra lineas que dan error
  funcione https://t.co/dmvaDSJV
grcpriest - 16 minutes ago
  Ya estoy agobiado , y me falta la e http://t.co/AtLHHCT
dani_latorre - 16 minutes ago
  ¿Donde anda la #agilidadnocturna? #aos2012
grcpriest - 19 minutes ago
  Ejercitando la lógica XD http://t.co/Kxq0MLI
eduo - 20 minutes ago
  No conocía la "Taza de 1 estrella" de Marco Arment - Genial - http://t.co/Te8G4IQb
tek_news - 24 minutes ago
  HNews: How LinkedIn betrayed a 5-man startup http://t.co/t3am6zTG #startup
grcpriest - 25 minutes ago
  Buenérrimo el Doodle de Google de hoy
democraciareal - 28 minutes ago
[INFO] => Configuration reloaded

```

When setting `box_around_status` to `false` you can specify a divider character that will be printed below the statuses.

Warning: The `box_around_status` option has precedence over `status_divider`

```

[styles]
box_around_status = false
status_divider = false
status_divider_char = -

```

This is how it looks like using `-` as a status divider:

```

dev social sys www tiled 0 Off 71% 23:2
15 tweets | 18 mentions
DvdBng - 3 minutes ago
http://t.co/Xo4T61dW Google construye una red neuronal de 16000 núcleos que es capaz de reco
haberla enseñado antes.

theom3ga - 3 minutes ago
De hecho me saqué una cuenta de un día de rapidshare para bajarme toda la colección de Andala

theom3ga - 3 minutes ago
Con eso de que ha muerto Juan Luis Galiardo, un vídeo a colación, que me encanta http://t.co/...

gregsidelnikov - 3 minutes ago
I speak of an idea. It is not said to anyone in particular. Like an ocean. It doesn't speak
particular but you can drown in it.

Vizzuality - 6 minutes ago
Forest monitoring Open Source tools that we developed last year highlighted on @google blog
http://t.co/YyNqpSHh

Ilgrim - 7 minutes ago
@theom3ga Hellraiser.

dialelo - 9 minutes ago
Writing docs for configuring turses, I should have done this long ago.

dabeaz - 9 minutes ago
And meanwhile, far far away from a mailing list, the LLVM backend to my compiler grows... IT
believe it works)

Capelare - 12 minutes ago
Me hace gracia la gente que se pregunta "para qué sirven" las cucarachas. ¿Y nosotros? ¿Para
humanos? Absurdo...

[INFO] => Configuration reloaded

```

Editor

You can also configure the position of the editor in the screen modifyng the `editor_horizontal_align` and `editor_vertical_align` options. The accepted values for this options are the following:

- `editor_horizontal_align`: left, center or right
- `editor_vertical_align`: top, middle or bottom

An example configuration with the editor positioned in the center of the screen:

```
[styles]
editor_horizontal_align = center
editor_vertical_align = middle
```

Here's how it looks like:


```
dev org social sys www tiled 0 On 185% 13:13
```

19 tweets 13 mentions 1 favorites alejandrosuarez - 5 minutes ago Debo cambiar urgente de banco. #lacaixa es una vergüenza desde hace meses... tb quitaré mis seguros de allí. Desastre total	MGGartz - 5 days ago @bindaloo @dialelo Oficialmente ya tengo el bachiller familia!	tek_news - 9 min HNNews: Talent s http://t.co/gFu
Juanje_Romero - 7 minutes ago Sentado en el meridiano de Greenwich	joe_di_castro - 7 days ago @dialelo Looks great! I'm testing it right now... and always fails when I tried with... YOU :) /cc @xekarfwtos	getpy - an hour "Tuples aren't they're for," b http://t.co/6N4
InfoUMA - 7 minutes ago "Jornada formativa de iniciación al Network #Marketing". Hoy a las 18h Facultad Ciencias Económicas y Empresariales. http://t.co/QCfyr2Cp	xekarfwtos - 8 days ago @dialelo @joe_di_castro I just replied on the issue thread on github. It is a great addition to #turses	CompSciFact - a Ten surprises f algebra http:// tek_news - a da
97 Tweet (twice enter key to validate or esc) >> For those of you who use turses: I've published 0.1.18 which is a bugfix release, please upgrade.		
InfoUMA - 11 minutes ago 21ª Tertulia- Taller: "Como implantar planes de formación en seguridad y salud en el trabajo". Mañana a las 12:15h http://t.co/Po8jPeqD	xekarfwtos - 9 days ago @dialelo Awesome is that I learn python, virtualenv, git, vim working on turses source code. Thank you.	http://t.co/geL tek_news - 8 da HNNews: Don't lo UX guy http://t
jonathanchacon - 11 minutes ago Preparando la próxima carga de trabajo para @Deliaprilis y @TyfAccSoft	joe_di_castro - 9 days ago @dialelo @xekarfwtos @nic0sphere Of course, all this, IMHO :)	CompSciFact - 8 It's the latenc http://t.co/i9u
clau5anj - 12 minutes ago Decidir quedarme en casa un día	joe_di_castro - 9 days ago @dialelo @xekarfwtos @nic0sphere And I think that height and width should be fit the 140 chars at start, would	tek_news - 8 da HNNews: Donald K http://t.co/fSr

Url format

You can choose how URLs are rendered tweaking the `url_format` option in the `styles` section. The following formats are available:

- **shortened:** URL shortened by Twitter (e.g. `t.co/foo`)
- **original:** Full original URL (e.g. `http://example.com/`)
- **display:** Original URL with an ellipsis if it's too long and trimming the protocol prefix (e.g. `example.com/a-very-long-`)

The default option is `display`. Here's an example of a configuration option to display the full original URLs:

```
[styles]
url_format = original
```


User info window

When you are focusing a status and press `i`, `turses` will show a popup with information about the author of the status. At the bottom of the information widget the last statuses posted by the user are shown.

The number of statuses to display is configurable via the `statuses_in_user_info` parameter. By default, it shows three statuses but we can easily change that to show the last five:

```
[styles]
statuses_in_user_info = 5
```

Debug

The last section of the configuration is `debug`, which is intended for developers.

```
[debug]
logging_level = 3
```


If you want to contribute to the project, this part of the documentation is for you.

Set up the development environment

You're free to setup up the environment in any way you like. Here is a way using virtualenv and virtualenvwrapper. If you don't have them, you can install them using:

```
$ pip install virtualenvwrapper
```

Virtual environments allow you to work on an installation of python which is not the one installed on your system. Especially, it will install the different projects under a different location.

To create the virtualenv environment, you have to do:

```
$ mkvirtualenv turses
```

Then you would have to install all the dependencies:

```
$ pip install -r requirements/dev.txt  
$ make install
```

Running the test suite

Each time you add a feature, there are two things to do regarding tests: checking that the tests run in a right way, and be sure that you add tests for the feature you are working on or the bug you're fixing.

The tests leaves under `/tests` and you can run them using `py.test`:

```
$ make test
```

Internals

Overview

Here is an overview of the multiple files that compound *turses*, each of them with a comment explaining their goal.

```
turses
- requirements
|   - base.txt
|   - dev.txt
- setup.py
- turses
  - api
  |   - base.py      # definition of an interface to the Twitter API
  |   - backends.py  # Twitter API implementations
  |   - debug.py     # mock API implementation for debugging
  |   - __init__.py
  - cli.py           # logic for launching `turses`
  - config.py        # configuration management
  - core.py          # core logic: controller and event handling
  - __init__.py
  - meta.py          # decorators and abstract base classes
  - models.py        # data structures
  - ui.py            # UI widgets
  - utils.py         # misc functions that don't fit elsewhere
```

turses.cli

Handle the invocation of *turses* from the command line.

```
turses.cli.main()
    Launch turses.
```

turses.config

class `turses.config.Configuration`

Generate and parse configuration files. When instantiated, it loads the defaults.

Calling `Configuration.parse_args()` with an `argparse.ArgumentParser` instance will modify the instance to match the options provided by the command line arguments.

Calling `turses.config.Configuration.load()` on this class' instances reads the preferences from the user configuration files. If no configuration or token files are found, this class will take care of creating them.

Offers backwards compatibility with the Tyrs configuration.

turses.core

This module contains the controller and key handling logic of *turses*.

class `turses.core.InputHandler(controller)`

Maps user input to calls to *Controller* functions.

class `turses.core.Controller(ui, api, timelines)`

The *Controller*.

turses.meta

This module contains abstract classes and decorators.

Decorators

`turses.meta.wrap_exceptions(func)`

Augments the function arguments with the *on_error* and *on_success* keyword arguments.

Executes the decorated function in a try except block and calls *on_success* (if given) if no exception was raised, otherwise calls *on_error* (if given).

`turses.meta.async(func)`

Decorator for executing a function in a separate `threading.Thread`.

`turses.meta.filter_result(func, filter_func=None)`

Decorator for filtering the output of *func* with *filter_func*.

Abstract base classes

The abstract base classes can be understood as interfaces; they define a set of methods and properties that their sub-classes must implement. They provide very general pieces of functionality.

class `turses.meta.ActiveList`

A list that contains an *active* element.

This abstract class implements some functions but the subclasses must define the `turses.meta.ActiveList.active` property, as well as `turses.meta.ActiveList.is_valid_index()` and `turses.meta.ActiveList.activate_last()` methods.

class `turses.meta.UnsortedActiveList`

A *ActiveList* in which the *active* element position can be shifted either to the beginning or to the end.

All the methods contained in this class are abstract.

class `turses.meta.Observable`

An implementation of the *observer* pattern.

Zero or more *observers* can *subscribe* to the changes in the instances of this class. When the instance changes, it will call its *notify* method, which loops through the *observers* and calls *update()* on them.

class `turses.meta.Updatable(update_function=None, update_function_args=None, update_function_kwargs=None)`

An abstract class for making a class *updatable*.

The constructor takes an update function and arguments used to update the subclasses of *Updatable*.

When *update()* is executed, *update_callback()* is called, passing it the result.

class `turses.meta.Observable`

An implementation of the *observer* pattern.

Zero or more *observers* can *subscribe* to the changes in the instances of this class. When the instance changes, it will call its *notify* method, which loops through the *observers* and calls *update()* on them.

class `turses.meta.Observer`

An abstract class that can subscribe to updates from *Observable* instances.

`turses.meta.notify(func)`

Wrap an instance method *func*, calling the instance's *notify* method after executing *func*.

turses.models

This module contains the data structures that power `turses` and the Twitter entities represented into it.

Base model

The model on which `turses` is based is `TimelineList`, a list of `Timeline` objects. This model is mapped into the list of buffers that appear on the user interface.

class `turses.models.TimelineList`

A list of `Timeline` instances that implements the `UnsortedActiveList` interface, thus having an *active* element and a group of adjacent *visible* timelines.

Twitter models

The Twitter entities represented on `turses` are the following:

class `turses.models.Timeline` (*name*='', *statuses*=None, ***kwargs*)

List of Twitter statuses ordered reversely by date, optionally with a name and a function that updates the current timeline and its arguments.

Its `Updatable` and implements the `ActiveList` interface.

class `turses.models.User` (*id*, *name*, *screen_name*, *description*, *url*, *created_at*, *friends_count*, *followers_count*, *favorites_count*, *status*=None)

A Twitter user.

class `turses.models.Status` (*id*, *created_at*, *user*, *text*, *author*='', *entities*=None, *is_reply*=False, *in_reply_to_user*='', *in_reply_to_status_id*=None, *is_retweet*=False, *retweeted_status*=None, *retweet_count*=0, *is_favorite*=False)

A Twitter status.

class `turses.models.DirectMessage` (*id*, *created_at*, *sender_screen_name*, *recipient_screen_name*, *text*, *entities*=None)

A Twitter direct message.

class `turses.models.List` (*id*, *owner*, *created_at*, *name*, *slug*, *description*, *member_count*, *subscriber_count*, *private*=False)

A Twitter list.

turses.ui

This module contains the curses UI widgets.

class `turses.ui.CursesInterface`

Creates a curses interface for the program, providing functions to draw all the components of the UI.

Provides a facade API to draw the representation of the `TimelineList`, help `HelpBuffer` and intro `Banner` screens.

Note: The list of widgets presented here is not complete.

Widget wrappers

turses make heavy usage of the `urwid.WidgetWrap` class to compose custom widgets.

`ScrollableWidgetWrap` helps to make widgets that implement the `Scrollable` interface, thus are navigable with the motion commands `up`, `down`, `scroll_to_top` and `scroll_to_bottom`.

class `turses.ui.ScrollableWidgetWrap` (*contents=None*)
A `urwid.WidgetWrap` for `Scrollable`, list-like widgets.

Twitter Widgets

Here's a list with some of the widgets that represent Twitter entities:

class `turses.ui.TimelinesBuffer` (*timelines=None, **kwargs*)
A widget that displays one or more *Timeline* objects.

Another widget can be placed on top of it.

class `turses.ui.StatusWidget` (*status*)
Widget containing a Twitter status.

class `turses.ui.UserInfo` (*user, last_statuses*)
A widget for displaying a Twitter user info.

Other widgets

class `turses.ui.HelpBuffer`
A widget that displays all the keybindings of the given configuration.

class `turses.ui.Banner`
Displays information about the program.

turses.utils

This module contains functions used across different modules.

Authors

Code contributors

- Nicolas Paris <<http://github.com/Nic0>>
- Alejandro Gómez <<http://github.com/alejandrogomez>>
- Sascha Kruse <<http://github.com/knopwob>>
- Poapfel <<http://github.com/poapfel>>
- Giannis Damigos <<http://github.com/xekarftos>>
- Joe di Castro <<http://github.com/joedicastro>>
- Paul Ivanov <<http://github.com/ivanov>>
- replicant <<http://github.com/replicant0wnz>>

- Wes Turner <<http://github.com/westurner>>
- Johannes Holmberg <<http://github.com/Jolmberg>>
- Marco Milanese <<http://github.com/kpanic>>
- Douglas La Rocca <<http://github.com/douglas-larocca>>

Active participants

Here's a list with the people that participate or have participated actively in making *turses* better:

- Diego Jose <<http://github.com/diegoxter>>
- Kismet <<http://github.com/kismet010>>
- Denise Tiersch <<http://github.com/dtiersch>>

Changelog

0.3.1

- Upgrade Tweepy dependency to 3.3
- Python 3 port by Douglas Larocca

0.3.0

- Upgrade Tweepy dependency to 3.1
- Drop oauthlib dependency

0.2.23

- Upgrade Tweepy dependency to 2.3

0.2.22

- Upgrade Tweepy dependency to 2.2

0.2.21

- Bugfix: Turses no longer crashes when clicking on the window

0.2.20

- Mouse scrolling support (thanks to Wes Turner)
- Bugfix: Escape HTML entities without borking tweet text (thanks to Denise Tiersch)

0.2.19

- Bugfix: Restore Python 2.6 compatibility

0.2.18

- Bugfix: use ASCII characters for reply and retweet indicators

0.2.17

- Make Ctrl-C act as escape
- Hashtag searches in sessions

0.2.16

- Update to tweepy 2.1, fix search functionality

0.2.15

- Configurable retweet and reply indicators

0.2.14

- Daemonize threads so they finish when quitting turses
- Fix a bug regarding SIGCONT handling

0.2.13

- *Y* as shortcut for retweeting and favoriting a tweet

0.2.12

- Upgrade to Tweepy 2.0 (hence Twitter API v1.1)

0.2.11

- Include `in_reply_to_status_id` parameter in replies

0.2.10

- Use HTTPS by default (thanks to Joe Di Castro)

0.2.9

- Configurable number of statuses in user info widget
- Sessions support
- Better information for SSL errors

0.2.8

- Python 2.6 support

0.2.7

- center focus when receiving tweets after scrolling down the bottom
- get your tweets that have been retweeted

0.2.6

- configurable URL format
- better browser integration (by Paul Ivanov)

0.2.5

- search commands also available with no timelines
- clear the status bar in every mode
- asynchronous search timeline addition
- keep columns when deleting a buffer within them
- bugfix: crashed when deleting all buffers

0.2.4

- bugfix: unable to focus the topmost status when having multiple columns

0.2.3

- per-account configuration
- bugfix: crashed when navigating to empty buffers

0.2.2

- fix regression: timelines not updating periodically

0.2.1

- bugfix: the Twitter entities were not processed every time

0.2.0

- logging

0.1.18

- bugfix: crash when rendering direct messages

0.1.17

- [developer docs](#)
- border around editor
- bugfix: help and version were removed by stdout replacement
- debug mode
- offline debugging

0.1.16

- show a popup with user info when pressing *i*

0.1.15

- configurable editor alignment
- make *turses* play nicely with terminal multiplexers (Joe Di Castro)
- follow and unfollow users typing their nick in an editor (Giannis Damigos)
- bugfix: pressing <Esc> in search user editor made *turses* crash
- bugfix: avoid duplicate usernames in replies

0.1.14

- bugfix: crash when shifting buffers to the beginning or end

0.1.13

- bugfix: could not remove own retweets
- bugfix: inconsistencies with help

0.1.12

- bugfix: missing key binding in help (Giannis Damigos)
- bugfix: status messages cleared the editor
- configurable status bar visibility
- changes to manual retweet template
- set console title to turses
- [docs on readthedocs](#)

0.1.11

- bugfix: exception when marking tweet as favorite

0.1.10

- expanded URLs for search results
- bugfix: crashed when expanded URLs were missing

0.1.9

- open focused status in a browser
- show expanded URLs

0.1.8

- bugfix: packaging error

0.1.7

- bugfix: inconsistencies when navigating tweets with arrow keys
- configurable status wrappers: box, divider or none

0.1.6

- colored urls
- colored favorites
- bugfix: non-ascii characters on templates made *turses* crash
- visual indicators for status types (retweet, reply, favorite)

0.1.5

- configurable tab text
- colored hashtags and usernames

0.1.4

- update all timelines periodically
- configurable default timelines
- bugfix: don't crash with empty timelines
- bugfix: manual retweet crashed
- bugfix: don't capture all input

0.1.3

- bugfix: packaging error

0.1.2

- bugfix: error with packaging

0.1.1

- bindings to update all timelines
- bugfix: *generate_token_file* instead of *create_token_file*

0.1.0

- binding to open focused status authors' tweets
- reload configuration
- configuration default location and format changed

0.0.15

- bugfix: DM recipient was not correctly resolved
- fetch newer tweets when scrolling up from the top and viceversa
- show retweets by default

0.0.14

- bugfix: logging

0.0.13

- thread view

0.0.12

- multiple visible timelines in columns
- fix bug with unicode input
- open URLs in browser

0.0.11

- include retweets in home timeline
- fix bug with non-existent method

0.0.10

- unread count
- mark all tweets in active timeline as read
- fix (again) a bug with mouse events

0.0.9

- compose tweet with same hashtags as the focused status
- create search timeline with hashtags from focused status

0.0.8

- fix bug: self follow/unfollow
- fix bug: editor signals
- direct messages :)
- persistent timeline cursor

0.0.7

- fix critical bug, missing dependency urwid

0.0.6

- fix bug with mouse events
- relative imports to avoid *ImportError* exceptions

0.0.5

- more colorful defaults
- see your own tweets
- search for a user's tweets

0.0.4

- follow and unfollow
- pluggable UI and API
- associate callbacks to API calls

0.0.3

- bug with non-ascii characters in search solved
- asynchronous API calls
- favorite/unfavorite tweets
- Favorites timeline

0.0.2

- tests with coverage check
- fixed bug with missing dependency in *setup.py*
- decoration for tabs

CHAPTER 4

Screenshots

```

dev social www tiled On 177% 8:4
|30|tweets|19|mentions|19|favorites|38|messages
pabloMP2P - 2 minutes ago
Jajajajaja, qué grandes! // cc @iaioflautas // RT
@veroniqueta82 Muy buena protesta de dos ancianos!
http://t.co/fxlyba7j

Guerraeterna & pabloMP2P - 3 minutes ago
Mossos infiltrados. Directos a la portada del Vogue.
http://t.co/VzDn9iTA

Timcast & pabloMP2P - 4 minutes ago
Aerial view of #m1gs http://t.co/Ms6Zpu6c

kix2902 - 6 minutes ago
En 5 minutos los testers de #realtimeWP tendran en su
mail la (espero) version definitiva de la aplicación ;)

fnichol & juanjeojeda - 6 minutes ago
Released v0.2.12 of user cookbook for #opschef w/ LWRP
notifies, @travisci/foodcritic, & experimental unit
tests! https://t.co/PCb4qMi1

pipotux & anaRB - 7 minutes ago
Me encantó meter cosas graciosas en los agradecimientos
de mis proyectos fin de carrera, y la tesis no iba a ser
menos http://t.co/p6qBM4As

pabloMP2P - 7 minutes ago
.@teleoperador Son los mossos infiltrados xDDDDDDDDDDDDDD

tek_news - 7 minutes ago
Reddit/p: A meta-howto tutorial: using the Squeak
[INFO] => Favorites fetched

nic0sphere - a day ago
New release of #Turses 0.1.6 with
ncurses twitter client. https://t.co/...
@dialelo

PepeGuer - 3 days ago
Mr. @dialelo we missed you last ni
http://t.co/tLDthltp

Capelare - 5 days ago
Genial el P**áez mandando un mensa
año pasado avisando de un cambio c
mañana... xD /@pepeguer @dialelo

aclark4life - 8 days ago
@dialelo Thanks! :-) That's one wa
better now.

Milouse - 11 days ago
As someone ask me yesterday, I use
Twitter client http://t.co/yikYqec
fine on linux or mac #www2012

python_majibu - 12 days ago
@dialelo Ahora turses es mi client
defecto en Ubuntu, muchas gracias
una maravilla y rapidisimo. @turse

python_majibu - 12 days ago
@dialelo Ya está solucionado el pr
instalar el paquete `python-dev` e
funciona de maravilla.

```

t

- `turses`, [1](#)
- `turses.cli`, [16](#)
- `turses.config`, [6](#)
- `turses.core`, [16](#)
- `turses.meta`, [17](#)
- `turses.models`, [18](#)
- `turses.session`, [6](#)
- `turses.ui`, [18](#)
- `turses.utils`, [19](#)

A

ActiveList (class in turses.meta), 17
async() (in module turses.meta), 17

B

Banner (class in turses.ui), 19

C

Configuration (class in turses.config), 16
Controller (class in turses.core), 16
CursesInterface (class in turses.ui), 18

D

DirectMessage (class in turses.models), 18

F

filter_result() (in module turses.meta), 17

H

HelpBuffer (class in turses.ui), 19

I

InputHandler (class in turses.core), 16

L

List (class in turses.models), 18

M

main() (in module turses.cli), 16

N

notify() (in module turses.meta), 17

O

Observable (class in turses.meta), 17
Observer (class in turses.meta), 17

S

ScrollableWidgetWrap (class in turses.ui), 19

Status (class in turses.models), 18
StatusWidget (class in turses.ui), 19

T

Timeline (class in turses.models), 18
TimelineList (class in turses.models), 18
TimelinesBuffer (class in turses.ui), 19
turses (module), 1
turses.cli (module), 16
turses.config (module), 6
turses.core (module), 16
turses.meta (module), 17
turses.models (module), 18
turses.session (module), 6
turses.ui (module), 18
turses.utils (module), 19

U

UnsortedActiveList (class in turses.meta), 17
Updatable (class in turses.meta), 17
User (class in turses.models), 18
UserInfo (class in turses.ui), 19

W

wrap_exceptions() (in module turses.meta), 17