
rogue_{scores}*Documentation*

Release 0.0.9

Baptiste Fontaine

October 14, 2014

1	Install	3
2	Guide	5
2.1	API Reference	5
2.2	Changes	7
	Python Module Index	9

`rogue_scores` is an online [Rogue](#) leaderboard as well as a script to upload local user scores to it. Once installed, you only have to run `rogue_scores` and the script will upload your scores for you.

The script will ask you which server to use the first time you'll start it and remember this choice. You can override the remembered choice by setting the `ROGUE_SCORES_SERVER` environment variable, or delete `~/ .rogue-scores-server` to remove the script's remembered choice.

Install

With `pip`:

```
[sudo] pip install rogue_scores
```


2.1 API Reference

2.1.1 `rogue_scores.scores`

This module is used to get and parse local user scores.

`rogue_scores.scores.get_scores (command='rogue')`

Return a list of local user scores, as given by the command `rogue -s`. These scores should be ordered from the best score to the worst one, and limited to 10, but this limit is not enforced by the function.

Keyword arguments:

- `command (string, default: rogue)`: the command to use to run `rogue`.

`rogue_scores.scores.parse_line (/)`

Parse a line from the scores given by `Rogue` and return a score as a tuple containing the user's name, score, and result. If the line wasn't correct, it returns `None`.

```
>>> parse_line(' 1    783 baptiste: killed on level 8 by a centaur.')
('baptiste', 783, 'killed on level 8 by a centaur.')
```

2.1.2 `rogue_scores.upload`

This module provides tools to upload `Rogue` scores to a remote server.

`rogue_scores.upload.post_scores (scores, **kwargs)`

Post some scores to a remote server and return a boolean depending on the request's success. These scores should be a list of tuples, as returned by `rogue_scores.scores.get_scores`. Optional keyword arguments are:

Keyword arguments:

- `protocol (string, default: http)`
- `target (string, default: localhost:5000)`

2.1.3 `rogue_scores.cli`

`rogue_scores.cli.run ()`

Run the command-line interface

`rogue_scores.cli.set_server()`

Ask the user for a server address and store it in a local file.

2.1.4 rogue_scores.web

rogue_scores.web.app

rogue_scores.web.store

This module provides functions to deal with scores stored on the Web server. Each score is represented as a dict-like object, with the following keys:

- `level`
- `score`
- `user`: user's name
- `cause`: the monster's name if the user was killed, the death cause if the user died of starvation or hypothermia
- `status (str)`: died, killed, quit

These keys can be `None` (or 0 for numerical ones) if the attribute is not set.

exception `rogue_scores.web.store.BadScoreFormatException(s)`

This exception is raised when a wrongly formatted score is given to a `ScoresStore` instance's `add` method.

class `rogue_scores.web.store.Score(**kwargs)`

A score. This object implements some dict-like methods to provide an easy access to its attributes. It have at least these ones: `level`, `score` (default: 0), `user`, `status`, `cause` (default: `None`).

New in version 0.0.7.

class `rogue_scores.web.store.ScoresStore(path=None, **kwargs)`

A scores store. This is based on a JSON file, but the interface should not depend on the underlying storage method.

```
>>> s = ScoresStore('/tmp/foo')
>>> s.add({'user': 'foo', 'level': 42, 'status': 'quit'})
1
>>> len(s)
1
>>> s.save()
None
```

New in version 0.0.7.

add (`*scs, **kwargs`)

Add one or more scores to the store. These are sanitized before insertion, and missing informations are added via parsing if possible. It returns the number of inserted items. An item is not added if:

- it's already present
- it doesn't contain basic info on the score, like no user or a null score

Keyword arguments can be used to set default values on all added scores.

get (`limit`)

Return at most `limit` scores

json (`**kwargs`)

Return a JSON representation of this store

save()

Save the current scores on disk, if the store's path is not None.

`rogue_scores.web.store.parse_text(text)`

Parse a score's text and return a dictionary of attributes: level, status, cause. Only a subset of these attributes might be returned if the text couldn't be parsed or some of them aren't relevant.

status can be either killed (by a monster), quit, died (e.g. of starvation) or won. This function should work with multiple text variants.

```
>>> parse_text("killed on level 12 by a quagga.")
{'level': 12, 'status': 'killed', 'cause': 'quagga'}
>>> parse_text("quit on level 3.")
{'level': 3, 'status': 'quit'}
>>> parse_text("died of hypothermia on level 6")
{'level': 6, 'status': 'died', 'cause': 'hypothermia'}
>>> parse_text("killed on level 3 by hypothermia.")
{'level': 3, 'status': 'died', 'cause': 'hypothermia'}
```

New in version 0.0.7.

rogue_scores.web.stats

This module helps computing interesting stats about the scores stored on the Web server.

`rogue_scores.web.stats.stats(scores)`

Compute stats on a ScoresStore, and return a dict that can then be used in a template. Each value is a string, and current keys are:

- `max_level`: higher level, with the corresponding user
- `most_active`: most games played by one user
- `best_killer`: monster with the most kills

If multiple users or monsters match a criteria for one of these keys, only one of them will be picked.

2.2 Changes

2.2.1 v0.0.9 (03/06/2014)

All changes of this version are in the upload script.

- support for `ROGUE_SCORES_SERVER`
- `--version` flag support
- redirections are now followed
- User-Agent header added

2.2.2 v0.0.8 (31/05/2014)

This release fixes a problem in the previous version where the limit of 20 scores has been removed.

2.2.3 v0.0.7 (31/05/2014)

This is a breaking change on the server side API, but is fully compatible with previous upload script versions. Unless you have Python code relying on `rogue_scores.web`'s functions, it's ok to upgrade.

- more log messages on the server side
- stats added on the Web page
- `/scores` now serves the JSON scores file, and `/scores?pretty` serves a pretty-printed version
- `rogue_scores.web` moved to `rogue_scores.web.app`
- scores-related functions moved from `web` to `web.store`, with `Score` and `ScoresStore` classes
- scores are now internally stored as JSON objects instead of list
- the Web page now shows each score's level, status and cause.

2.2.4 v0.0.6 (27/05/2014)

- `ROGUE_SCORES_PATH` support
- the default Web scores file location is now the current directory

2.2.5 v0.0.5 (27/05/2014)

- `http[s]` is not mandatory anymore when entering the remote server
- help text added on the Web index page

2.2.6 v0.0.4 (26/05/2014)

- local server name file parsing fixed

2.2.7 v0.0.3 (26/05/2014)

- default Web scores file location moved to `~/ .rogue-scores.json`
- logging added

2.2.8 v0.0.2 (26/05/2014)

- dependencies list fixed

2.2.9 v0.0.1 (26/05/2014)

- initial release

r

- `rogue_scores.cli`, 5
- `rogue_scores.scores`, 5
- `rogue_scores.upload`, 5
- `rogue_scores.web`, 6
- `rogue_scores.web.stats`, 7
- `rogue_scores.web.store`, 6

A

`add()` (`rogue_scores.web.store.ScoresStore` method), 6

B

`BadScoreFormatException`, 6

G

`get()` (`rogue_scores.web.store.ScoresStore` method), 6

`get_scores()` (in module `rogue_scores.scores`), 5

J

`json()` (`rogue_scores.web.store.ScoresStore` method), 6

P

`parse_line()` (in module `rogue_scores.scores`), 5

`parse_text()` (in module `rogue_scores.web.store`), 7

`post_scores()` (in module `rogue_scores.upload`), 5

R

`rogue_scores.cli` (module), 5

`rogue_scores.scores` (module), 5

`rogue_scores.upload` (module), 5

`rogue_scores.web` (module), 6

`rogue_scores.web.stats` (module), 7

`rogue_scores.web.store` (module), 6

`run()` (in module `rogue_scores.cli`), 5

S

`save()` (`rogue_scores.web.store.ScoresStore` method), 6

`Score` (class in `rogue_scores.web.store`), 6

`ScoresStore` (class in `rogue_scores.web.store`), 6

`set_server()` (in module `rogue_scores.cli`), 5

`stats()` (in module `rogue_scores.web.stats`), 7