copydog Documentation

Release 0.1

coagulant

August 25, 2014

Contents

1 Installation **3**

2 Configuration **5**
2.1 Trello . . . 5
2.2 Redmine . . . 5
2.3 Example config . . . 5

3 Running copydog **7**
3.1 Deamon . . . 7

4 How it works **9**
4.1 Fields mapping . . . 9
4.2 Markup conversion . . . 9
4.3 Storage . . . 10

5 Development **11**
5.1 Launching tests . . . 11
5.2 REST API references . . . 11

6 Changelog **13**
6.1 ver 0.1 (2012-08-18) . . . 13

Copydog converts issues between Redmine and Trello on the fly. It's a small third-party daemon, monitoring changes in both systems and keeping them in sync as much as possible.

You might find this tool useful after reading How it works.

Warning: Copydog is in active development, it's not ready for production yet. However, please, feel free to experiment/test and use the code.

CHAPTER 1

Installation

Copydog is available via PyPI:

```
pip install copydog
```

You also need a Redis instance to store intermediate results of syncronization. Follow installation instructions.

For best experience please install Pandoc to convert issues descriptions between Markdown and Textile. However, it's optional.

Copydog runs on python 2.6-2.7 only. Python 3 support is planned in future releases.

CHAPTER 2

Configuration

For copydog to work you need both Trello and Redmine accounts.

2.1 Trello

If you don't have an account, head over to signup page and register. After that you need to obtain you API key and user token.

- Generate the API key for your application (need to be logged in).
- A token grants permissions (read, write or both) to one application for a specific user. To get a token for yourself, visit the docs.

Note: Obtain Trello token granting forever read and write access:

`https://trello.com/1/authorize?key=substitutewithyourapplicationkey&name=My+Application&expiration=ne`

2.2 Redmine

Copydog works with Redmine 1.3 and higher. You'll be needing your redmine instance url and your API key.

To obtain the API key visit your profile page at redmine `http://exampleredminehost/my/account` and click `Show link` under `API access key` section.

2.3 Example config

Copydog reads config variables from yaml files, so we need to make one. Here is how config file might look like:

```
clients:
  redmine:
    host: http://redmine.org
    api_key: ac7785e2c593ad6d7f539f2f90be26ba0851d18a
    project_id: playground
    write: 1
  trello:
    api_key: 9e90d281ed678b56b041871c3651ee2d
    token: e80a1138e0b5ea08d506fcabe6c17196542af7ee684c6c24b6f5b79
```

```
    board_id: 4fe889e4c23b476f4a189ca5
    write: 1
storage:
  redis:
    host: localhost
    port: 6379
    db: 0
    password: null
copydog:
  pandoc: '/usr/bin/pandoc'
  beat: 60
```

Note, how config is separated into sections: `clients`, `storage` and `copydog`. First one is required, other are completely optional. Clients have to be `redmine` and `trello`, with following attributes:

- **`redmine`**
  - `host` - the host of your redmine instance
  - `api_key` - your API key to access Redmine API
- **`trello`**
  - `api_key` - the API key of your Trello app
  - `token` - your consumer token to access Trello API

This keys can be obtained via browser, please read above sections for both *Redmine* and *Trello*.

Note: Copydog supports syncing one redmine project with one trello board at a time, so you need to specify `project_id` (string slug or integer) and `board_id` (string id).

Storage config might be completely omitted if you're using default Redis connection. Write flag allows copydog to modify contents on a client, set it to 0 to disable sync writes to either redmine or trello.

You can optionally provide `tracker_id` and/or `fixed_version_id` in redmine section to limit the number of issues being synced.

Note: While Redmine can handle thousands of issues painlessly, Trello is simply not suited for that amount of cards per board. I recommend using `tracker_id` or `fixed_version_id` filters to make better use of Trello.

By default copydog polls servers every minute. Is not a big burden for external APIs and sufficient to stay up to date. If it doesn't suite your needs, feel free to change the beat frequency under `copydog` section in config (it's called `beat` and is measured in seconds between polls.

CHAPTER 3

Running copydog

To launch the app:

```
copydog --config=<path_to_your_yaml_config>
```

Copydog will start monitoring *new* changes in both services and mirror them accordingly. If you wish to sync all existing issues/card, use `--fullsync` option:

```
copydog --fullsync --config=<path_to_your_yaml_config>
```

3.1 Deamon

If you're not developing copydog it's useful to run it as daemon process. To daemonize copydog, run it with a `start` argument:

```
copydog start --config=<path_to_your_yaml_config>
```

Copydog will run in background unless you stop it:

```
copydog stop --config=<path_to_your_yaml_config>
```

CHAPTER 4

How it works

Copydog polls both Redmine and Trello in turns, converting data from one service to the other. It queries first service for issues, updated since the last read and saves their identifiers and timestamps in storage. If there are any, they're converted into sister service type. Copydog tracks both new issues/cards and updates of existing ones by storing references between issues and cards. Trello cards are created with comments, featuring urls to corresponding redmine issues.

4.1 Fields mapping

Copydog tries to be smart when transferring cards to issues and vice versa. Redmine statuses are associated with Trello lists and are mapped by exact name match, so make sure you have same set of Statuses and Lists in your project and your board. Assigned members are linked by username or full name as a fallback.

Redmine	Trello	Comment
subject	name	
description	desc	Text is converted with pandoc, if available. See *Markup conversion*.
assigned_to	idMembers	Redmine doesn't support multiple assignees, the first one is taken.
status_id	idList	Copydog maps each status to list by name.
project_id	board_id	One board is synced with one project only.
due_date	due	

Other data like priorities, comments, labels are not synced.

4.2 Markup conversion

Copydog tries to use Pandoc tool to convert issue text between services. For example, Trello understands Markdown and Redmine uses Textile. If you don't have pandoc installed, texts will be transferred as is. It doesn't always look nice, so I advise you to install pandoc anyway.

You should provide a path to pandoc binary in config under `copydog` section to make it work.

```
copydog:
  pandoc: '/usr/bin/pandoc'
```

4.3 Storage

Copydog needs intermediate storage to save references between issues in Redmine and Trello. It also save datetime when items were last updated to make sure we're not going to sync issues back and forth forever. Copydog remembers time of last sync, so it will resume its work from the same spot.

Redis database is used for storing this data. If you wish to use another storage, you should write your own backend.

CHAPTER 5

Development

Copydog is developed and maintained by Baryshev Ilya. Feel free to submit issues or comments at development Trello board.

5.1 Launching tests

To launch tests execute:

```
nosetests
```

Some tests make actual API read requests, but they're disabled by default, to run them use:

```
nosetests -c all
```

These tests will pass if you have following env variables set:

- `COPYDOG_REDMINE_HOST` - the host of your redmine instance
- `COPYDOG_REDMINE_API_KEY` - your API key to access Redmine API
- `COPYDOG_TRELLO_API_KEY` - the API key of your Trello app
- `COPYDOG_TRELLO_TOKEN` - your consumer token to access Trello API

5.2 REST API references

- Redmine API docs
- Trello API overview
- Trello API docs

CHAPTER 6

Changelog

6.1 ver 0.1 (2012-08-18)

- Initial release