
tcp Documentation

Release dev

Mathieu Agopian

September 21, 2012

CONTENTS

1	Installation	3
2	Deployment	5
2.1	Manual usage	5
2.2	API usage	6
2.3	Administration	6

TCP is a web REST service, that you can use to update and validate video embed codes from video providers (think Youtube, Dailymotion, Vimeo...).

The source code is hosted on [github](#).

INSTALLATION

Getting the code:

```
git clone git@github.com:magopian/tcp.git
cd tcp
virtualenv -p python2 env
source env/bin/activate
add2virtualenv .
pip install -r requirements.txt
```


DEPLOYMENT

Create `tcp/settings.py` and put the minimal stuff in it:

```
from default_settings import *

ADMINS = ( ('Your name', 'email@example.com'),
)
MANAGERS = ADMINS

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql_psycopg2',
        'NAME': 'tcp',
        'USER': 'postgres',
    },
}

SECRET_KEY = 'something secret'

TIME_ZONE = 'Europe/Paris'

EMAIL_HOST = 'mail.your_domain.com'
EMAIL_SUBJECT_PREFIX = '[TCP] '
```

Then deploy your application and run `django-admin.py syncdb --settings=tcp.settings` to create the database tables, then `django-admin.py migrate --settings=tcp.settings` to run all `South` migrations.

The very first time you launch the application, once the database is set up, you may want to load some example providers and matches. To do that, use the following command:

```
django-admin.py loaddata providers_matches.json --settings=tcp.settings
```

Be aware that if you do that again later, it will simply overwrite the entries, loosing your modifications to those entries, if any.

2.1 Manual usage

Launch your web browser on your server's root url, or check out the demo available on <http://tcp.agopian.info>.

There's a textarea, put your video embed code in there, and watch the magic happen:

- it first tries to find the provider (Youtube, Google Video, Dailymotion, Vimeo and CrossTV are already set up and configured)

- it then updates the embed code to the one configured in the administration
- it finally validates if the video is still accessible (if it finds anything lower than a 400 HTTP status code)

Examples:

Dailymotion:

```
<object width="480" height="381"><param name="movie" value="http://www.dailymotion.com/swf/k6Lg9UXest"
```

Vimeo:

```
<iframe src="http://player.vimeo.com/video/17921737?title=0&byline=0&portrait=0&color=ffffff" width="
```

Youtube:

```
<object width="480" height="390"><param name="movie" value="http://www.youtube.com/v/WjsTx0RL"
```

2.2 API usage

Setting your `Accept-encoding` header to `application/json` will trigger the “API mode”. It’ll answer with json data.

Example:

```
$ curl -H "Accept-encoding: application/json" --data-urlencode 'initial_code=<iframe src="http://play
{"video_link": "http://player.vimeo.com/video/17921737", "is_valid": "true", "clean_code": "<iframe s
```

The json contains the following fields:

- `video_link`: the canonical link of the video to be embedded
- `is_valid`: if a `HEAD` request on the validation link configured in the administration returns a HTTP status code less than 400.
- `clean_code`: the new embed code
- `message`: optional information message
- `provider`: the video provider

2.3 Administration

The administration is available at the `/admin` url, or on the demo deployed at <http://tcp.agopian.info/admin>. You may login using the demo credentials `admin:admin`.

There’s two items available:

- Requests: the list of all requests (manual and through the API)
- Providers: where the configuration of new (or existing) providers is done

2.3.1 Providers configuration

The <http://tcp.agopian.info/admin/provider/provider/> link displays the list of current providers. It’s possible to search a provider by its name using the top search bar.

Adding or modifying a provider is as simple as clicking on the “Add provider” button in the top-right corner, or on the name of an existing provider.

- Name: purely informal
- Link template: template used to build the canonical link from the video id. It should contain the `{{ video_id }}` template tag where the video id should be replaced.
- Embed template: template used to build the new embed code. The link built from the previous template will replace the `{{ video_link }}` tag.
- Validation link template: used to validate that the video is still available (returns a `HTTP status code` less than 400).

2.3.2 Link Matches

Those are the [regular expressions](#) used to map an embed code to a provider. It should capture the video id, used in the link template above.

Please take example on the current configured providers!