
Nuancier Documentation

Release

Pierre-Yves Chibon <pingou@pingoured.fr>

December 17, 2013

Contents

1	Deployment	3
1.1	From sources	3
1.2	From system-wide packages	3
1.3	Set-up WSGI	4
1.4	For testing	4
2	Configuration	5
2.1	The secret key	5
2.2	The database URL	5
2.3	The admin group	5
2.4	The pictures folder	6
2.5	The cache folder	6
2.6	The thumb size	6
3	Administrate	7
3.1	Users	7
3.2	Upload new pictures for an election	7
3.3	Administration panel	7
3.4	Create elections	8
3.5	Open/Close election for votes	8
3.6	Publish results of an election	8
3.7	Generate cache	8
4	Usage	9
4.1	The page organisation	9
4.2	Finding the elections open	9
4.3	Vote	9
4.4	Finding the elections published	10
4.5	Consulting the results of an election	11
4.6	Statistics of an election	11
5	Development	13
5.1	Get the sources	13
5.2	Dependencies	13

5.3	Run nuancier for development	13
5.4	Coding standards	14
5.5	Send patch	14
5.6	Unit-tests	14
5.7	Database changes	15
6	Contributing	17
7	About nuancier	19
7.1	Templates	19
7.2	Lightbox2	19
7.3	Icons	19
7.4	Contributors	19
8	Indices and tables	21

Nuancier is a web-based voting application for the supplementary wallpapers of Fedora.

For each release of Fedora a number of supplementary wallpapers are provided. These wallpapers are submitted by contributors to the community who choose among them those that will be included in the release.

The current election application does not support voting on images. Nuancier aims at providing a simple application uploading and voting on these images/candidates.

Nuancier is meant as a temporary solution before elections or nuancier takes over, but in the meanwhile it does provide a nicer way to vote for the supplementary wallpapers.

Resources:

- [Home page](#)
- [Documentation](#)
- [Git repository](#)

Contents:

Deployment

1.1 From sources

Clone the source:

```
git clone https://github.com/fedora-infra/nuancier.git
```

Copy the configuration files:

```
cp nuancier.cfg.sample nuancier.cfg
```

Adjust the configuration files (secret key, database URL, admin group...). See *Configuration* for detailed information about the configuration.

Create the database scheme:

```
NUANCIER_CONFIG=/path/to/nuancier.cfg python createdb.py
```

Set up the WSGI as described below.

1.2 From system-wide packages

Start by install nuancier:

```
yum install nuancier
```

Adjust the configuration files: `/etc/nuancier/nuancier.cfg`. See *Configuration* for detailed information about the configuration.

Create the database scheme:

```
NUANCIER_CONFIG=/etc/nuancier/nuancier.cfg python /usr/share/nuancier/nuancier_createdb.py
```

Set up the WSGI as described below.

1.3 Set-up WSGI

Start by installing `mod_wsgi`:

```
yum install mod_wsgi
```

Then configure apache:

```
sudo vim /etc/httpd/conf.d/nuancier.conf
```

uncomment the content of the file and adjust as desired.

Then edit the file `/usr/share/nuancier/nuancier.wsgi` and adjust as needed.

Then restart apache and you should be able to access the website on <http://localhost/nuancier>

Note: Flask provides also some documentation on how to [deploy Flask application with WSGI and apache](#).

1.4 For testing

See *Development* if you want to run nuancier just to test it.

Configuration

There are the main configuration options to set to have nuancier running. These options are all present and described in the `nuancier.cfg` file.

2.1 The secret key

Set in the configuration file under the key `SECRET_KEY`, this is a unique, random string which is used by Flask to generate the `CSRF` key unique for each user.

You can easily generate one using `pwgen` for example to generate a 50 characters long random key

```
pwgen 50
```

2.2 The database URL

Nuancier uses `SQLAlchemy` has Object Relationship Mapper and thus to connect to the database. You need to provide under the key `DB_URL` in the configuration file the required information to connect to the database.

Examples URLs are:

```
DB_URL=mysql://user:pass@host/db_name
DB_URL=postgres://user:pass@host/db_name
DB_URL=sqlite:////full/path/to/database.sqlite
```

Note: The key `sqlalchemy.url` of the `alembic.ini` file should have the same value as the `DB_URL` described here.

2.3 The admin group

Nuancier relies on a group of administrator to create new elections, open or close them for voting and open or close the publication of the results and (re)generate the cache. The `ADMIN_GROUP` field in the configuration file refers to the `FAS` group that manages this nuancier instance.

See *Usage* for details explanations on the different administration layer of nuancier.

Note: Several groups of administrators can be set using either () or [].

2.4 The pictures folder

The `PICTURE_FOLDER` field provides to the application the full path to the folder in which are placed the pictures candidates for the elections (within a folder, specific for each election).

2.5 The cache folder

The `CACHE_FOLDER` field provides to the application the full path to the folder in which the application is allowed to generate the thumbnails of the pictures present in the `PICTURE_FOLDER`.

Note: This folder should be writable by the application (ie: apache).

2.6 The thumb size

In order to decrease the weight of the pages displaying all the pictures candidate to an election, nuancier creates thumbnails of these pictures. These thumbnails are generated with anti-aliases to maintain a certain quality.

The `THUMB_SIZE` is a set of length, width coordinate providing indication to nuancier about the desired size of the thumbnails.

By default `THUMB_SIZE` is at 256x256.

Administrate

3.1 Users

Nuancier has basically two levels for the users:

- administrators
- users

3.1.1 Administrators

Administrators are people with an account on the [Fedora account system \(FAS\)](#) and belong the one of administrator groups as set in the *Configuration*.

Administrators are the only people allowed to create an election, open or close it for votes, open or close the results and generate the cache (thumbnails).

3.1.2 Users

Users are people with an account on the [Fedora account system \(FAS\)](#) and belong to at least one more group than the `fedora_cla` group which every contributor should sign to contribute to Fedora.

3.2 Upload new pictures for an election

When preparing an election, the election wrangler needs to gather all the candidate wallpapers into a folder, with a unique name and place this folder in the directory specified under `PICTURE_FOLDER` in *Configuration*.

3.3 Administration panel

After logging in, if you are in the administrator group, you will see an `Admin` entry in the menu.

If you click on this `Admin` link you will arrive to the index page of the administration panel.

This page shows you all elections registered with their information and for each if they are open for vote or not and if their results are public or not. It offers the possibility to (re-)generate the cache for an election and once the election is closed, a link to some statistics about it.

3.4 Create elections

Click on the link `Create a new election` from the administration panel.

The form to will ask for:

- `election name`: the name of the election, this will be used as link throughout the application. Example name might be : *Fedora 20 wallpaper*
- `Name of the folder containing the pictures`: this specifies the name of the folder containing the pictures for that election that has been placed in the folder specified under `PICTURE_FOLDER` in *Configuration*. .. note:: It is a good idea to keep this name simple, unique, ascii and
and without spaces.
- `Year`: the year the election is taking place, this is purely for information.
- `Open`: This is a checkbox to specify whether this election is already open for votes or not.
- `URL to claim a badge for voting`: this allows to specify a link where people will be able to go to collect a badge announcing they participated on this election. You should coordinate with the people of the *badge* project to get this link.
- `Number of votes a user can make`: this specifies the number of choices a user can make for this election. For example, a user might be allowed to select only 16 wallpapers, thus this field should be *16*.
- `Generate cache`: this is checkbox offering to generate the cache assuming the pictures have already been placed on the `PICTURE_FOLDER`, together with the `infos.txt` file.

3.5 Open/Close election for votes

Once an election is opened for vote or has ended, the administrator can simply log in nuancier, go to the administration panel, find the correct election and on the *Open* column click on the `toggle` link.

If fedmsg is installed on the server, fedmsg messages are published for these events.

3.6 Publish results of an election

Once an election has ended, to publish its results, the administrator can simply log in nuancier, go to the administration panel, find the correct election and on the *Published* column click on the `toggle` link.

If fedmsg is installed on the server, fedmsg messages are published for these events.

3.7 Generate cache

To decrease the weight of the page where all the candidates of an election are shown, nuancier generates thumbnails.

To generate the cache of an election, the administrator needs to log in nuancier, go to the administration panel, find the correct election and click on the `(Re-)generate cache`.

Usage

4.1 The page organisation

At the top of the page three tabs are presented.

- The `Home` tab brings you back to the front page of the application where is some text explaining what this application is about.
- The `Elections` tab presents you the list of all the election present in the database, with their name, year and status.
- The `Results` tab gives you a list of links pointing to the result page of each election whose results have been published.

4.2 Finding the elections open

To find all the election open, go to the `Elections` tab at the top of the page.

In the column `Open` of the table the election open for vote are marked with:



The elections whose votes are either not yet opened or already closed are marked with the symbol:



4.3 Vote

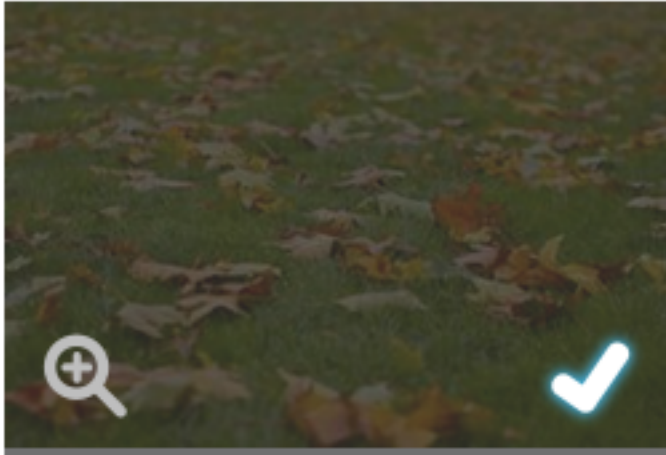
If an election is opened, it is possible to vote on it.

Go to the `Elections` tab at the top of the page and click on the name of an election which has the following symbol in the `Open` column of the table:



This will bring you to the page to vote.

In front of you, in the middle of the page, on a white background are presented all the candidates for the election you choosed. By hovering your mouse over the image, you can see two icons appearing:



On the bottom left is a *google* icon.



If you click on the *google* icon, you will be able to see a bigger version of the image, allowing to see it in more detail.

On the bottom right is a *check* icon.



If you click on this *checked* icon, you will select this image to vote on.

As you select candidates, these candidates will disappear from the central list of candidates and move to the grey column on the right where are presented all the candidates you selected.

At the top of this selection, still in the grey column on the right is displayed the number of candidates you selected versus the maximum number of candidates you are allowed to select.

Once you selection is made, simply click on `Submit Votes` on the grey column on the right.

You will be brought back to the list of elections and thanks for voting.

Note: You may vote on as many or as little candidates as you like up to the maximum allowed per elections. If you select less than the maximum number of candidates you will be able to come back on the election page and complete your vote (thus allowing to vote in several times).

Warning: You cannot change your vote once you have submitted it.

4.4 Finding the elections published

By published, is meant election whose results have been published, thus elections whose results are public.

To find all the election whose results are public, go to the `Elections` tab at the top of the page.

In the column `Published` of the table the election whose results are public are marked with:



While elections whose results are not yet public are marked with the symbol:



4.5 Consulting the results of an election

You can find the list of all the elections whose results are published by clicking on the tab `Results` at the top of the page.

There you can click on the election of your interest and access the result page.

The result page will show you a table with for all the candidates, their name, number of votes and thumbnails ordered by decreasing number of votes.

4.6 Statistics of an election

You can find the list of all the elections whose results are published by clicking on the tab `Results` at the top of the page.

There you can click on the `stats` link of the election of interest.

This page displays some statistics about the election : - Number of participants - Number of votes, - Maximum number of vote per person - Bar graph indicating how many people voted on how many candidates, for

example: 3 person voted to 4 candidates while 10 voted only on 2 candidates.

Development

5.1 Get the sources

Anonymous:

```
git clone https://github.com/fedora-infra/nuancier.git
```

Contributors:

```
git clone git@github.com:fedora-infra/nuancier.git
```

5.2 Dependencies

The dependencies of nuancier are listed in the file `requirements.txt` at the top level of the sources.

5.3 Run nuancier for development

Create the database scheme:

```
python createdb.py
```

Run the server:

```
./runserver
```

You should be able to access the server at <http://localhost:5000>

Every time you save a file, the project will be automatically restarted so you can see your change immediatly.

Note: You may want to adjust the values in `nuancier/default_config.py` especially the `ADMIN_GROUP` if you are not part of the default one. See *Configuration* for more information about the configuration.

5.4 Coding standards

We are trying to make the code [PEP8-compliant](#). There is a [pep8 tool](#) that can automatically check your source.

We are also inspecting the code using [pylint](#) and aim of course for a 10/10 code (but it is an asymptotic goal).

Note: both pep8 and pylint are available in Fedora via yum:

```
yum install python-pep8 pylint
```

5.5 Send patch

The easiest way to work on nuancier is to make your own branch in git, make your changes to this branch, commit whenever you want, rebase on master, whenever you need and when you are done, send the patch either by email or via the github pull-request mechanism.

The workflow would therefore be something like:

```
git branch <my_shiny_feature>
git checkout <my_shiny_feature>
<work>
git commit file1 file2
<more work>
git commit file3 file4
git checkout master
git pull
git checkout <my_shiny_feature>
git rebase master
git format-patch -2
```

This will create two patch files that you can send by email to submit in the trac.

5.6 Unit-tests

Nuancier has a number of unit-tests providing at the moment a full coverage of the backend library (nuancier.lib).

We aim at having a full (100%) coverage of the whole code (including the Flask application) and of course a smart coverage as in we want to check that the functions work the way we want but also that they fail when we expect it and the way we expect it.

Tests checking that function are failing when/how we want are as important as tests checking they work the way they are intended to.

`runtests.sh`, located at the top of the sources, helps to run the unit-tests of the project with coverage information using [python-nose](#).

Note: You can specify additional arguments to the nose command used in this script by just passing arguments to the script.

For example you can specify the `-x / --stop` argument: *Stop running tests after the first error or failure* by just doing

```
./runtests.sh --stop
```

Each unit-tests files (located under `tests/`) can be called by alone, allowing easier debugging of the tests. For example:

```
python tests/test_model.py
```

Similarly as for nose you can also ask that the unit-test stop at the first error or failure. For example, the command could be:

```
python -m unittest -f -v tests.test_model
```

Note: In order to have coverage information you might have to install `python-coverage`

```
yum install python-coverage
```

5.7 Database changes

We try to make the database schema as stable as possible, however once in a while we need to change it to add new features or information.

When database changes are made, they should have the corresponding change handled via [alembic](#).

See the [alembic tutorial](#) for complete information on how to make a revision to the database schema.

The basic idea is to create a revision using (in the top folder):

```
alembic revision -m "<description of the change>"
```

Then edit the file generated in `alembic/versions/` to add the correct command for upgrade and downgrade (for example: `op.add_column`, `op.drop_column`, `op.create_table`, `op.drop_table`).

Contributing

If you're submitting patches to nuancier, please observe the following:

- Check that your python code is [PEP8-compliant](#). There is a [pep8](#) tool that can automatically check your source.
- Check your code quality using [pylint](#).
- Check that your code doesn't break the test suite. The test suite can be run using the `runtests.sh` shell script at the top of the sources. See [Development](#) for more information about the test suite.
- If you are adding new code, please write tests for them in `tests/`, the `runtests.sh` script will help you to see the coverage of your code in unit-tests.
- If your change warrants a modification to the docs in `doc/` or any docstrings in `nuancier/` please make that modification.

Note: You have a doubt, you don't know how to do something, you have an idea but don't know how to implement it, you just have something bugging you?

Come to see us on IRC: `#fedora-apps` on irc.freenode.net or via its [github tracker](#).

About nuancier

Nuancier is a web-based voting application for the supplementary wallpapers of Fedora. Here are some more information about the technology, layout and artwork used in this application.

7.1 Templates

Nuancier re-used the layout of [Koji](#) adjusted as desired.
(License: LGPLv2)

7.2 Lightbox2

Nuancier uses [lightbox2](#) as javascript library to pop-up the pictures and present a larger version.
(License: CC-BY)

7.3 Icons

The icons used are coming from the [echo icon theme](#)
(License: CC-BY-SA)

7.4 Contributors

Nuancier is the results of a team effort started by Jenn and Gnookii and followed by a number of contributors. It was first named `nuancier-lite` and renamed to `nuancier` when support for uploads and moderation was added.

On September 23 2013, the list looks as follow:

Number of commits	Contributor
93	Pierre-Yves Chibon <pingou@pingoured.fr>
18	Ralph Bean <rbean@redhat.com>
9	Ryan Lerch <ryanlerch@fedoraproject.org>

This list is generated using

```
git shortlog -s -n -e
```

Indices and tables

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