opps Documentation

Release 0.2.5

Thiago Avelino

June 18, 2014

Contents

An *Open Source Content Management* for the **magazine** websites and **high-traffic**, using the Django Framework.

Topics

CHAPTER 1

Contributing to Opps

As an open source project, Opps welcomes contributions of many forms.

Examples of contributions include:

- Code patches
- Documentation improvements
- Bug reports and patch reviews
- Write CHANGELOG

Warning: pull requests are ignored! File a ticket to suggest changes.

CHAPTER 2

Contacts

The place to create issues is opps's github issues. The more information you send about an issue, the greater the chance it will get fixed fast.

If you are not sure about something, have a doubt or feedback, or just want to ask for a feature, feel free to join our mailing list, or, if you're on FreeNode (IRC), you can join the chat #opps.

CHAPTER 3

Run example

Download and install Opps

```
git clone https://github.com/opps/opps.git
cd opps
python setup.py develop
```

Now you can start a new Opps project

```
opps-admin.py startproject PROJECT_NAME
cd PROJECT_NAME
python manage.py syncdb --noinput
python manage.py migrate
python manage.py collectstatic --noinput
python manage.py runserver
```

CHAPTER 4

Tests

To run the test you must have an Redis instance running locally on the 6379 port (default one). Then, just type the following

```
make test
```

To run tests in multiple python versions, first install tox and then run the tox command:

```
pip install tox
tox
```

4.1 VirtualBox

```
vagrant box add opps http://mirror.oppsproject.org/opps.box
vagrant up
vagrant ssh
workon opps
cd /home/vagrant/opps/example/
python manage.py runserver 0.0.0.0:8000
```

CHAPTER 5

Sponsor

- YACOWS
- Digital Ocean

CHAPTER 6

License

opps is licensed under the MIT License

CHAPTER 7

Mailing list

CHAPTER 8

Quick links

8.1 Overview

8.1.1 Dependencies

Opps makes use of as few libraries as possible (apart from a standard Django environment), with the following dependencies

- Python 2.7
- Django >= 1.5
- Python Imaging Library - for image resizing
- South - for database migrations
- Django Taggit
- Django Mptt

8.1.2 Features

- Write in Django
- Containar manager (**container** is content type in *Opps CMS*)
- – Save draft and and preview
- – Dynamic custom field, add field in specific container
- WYSIWYG editing (more one option)
- Container Box manager (Custom channel home page, add dynamic/fix box)
- Channel organize via tree (via mptt, not level limit)
- Media file manager, default manager images
- – Multi upload
- User permission in Admin, manager site access on admin
- Configurable dashboard (used grappelli admin theme)
- API for custom container types
- Search engine

- Multi-Site
- SEO friendly URLs and meta data
- Mobile detect
- JVM compatible (via Jython)
- .NET Framework compatible (via IronPython)

8.1.3 Browser Support

Opps's admin (used Django Grappelli) interface works with all modern browsers.

8.1.4 Language Translations

Opps makes full use of translation strings, which allow Opps to be translated into multiple languages using Django's internationalization methodology. Translations are managed on the Transiflex website but can also be submitted via GitHub. Consult the documentation for Django's internationalization methodology for more information on creating translations and using them.

8.1.5 Sites Using Opps

- Portal Virgula
- Portal Guiame
- Grupo Troiano
- Troiano
- Brands and Values
- Brigh Thouse
- Brand Insights
- JovemPan AM
- JovemPan FM

8.2 Getting Started

8.2.1 Getting help

Should you run into trouble and can't figure out how to solve it yourself, you can get help from either our mailinglist or IRC channel #opps on the irc.freenode.net network.

8.2.2 Start

```
git clone https://github.com/opps/opps.git
cd opps
python setup.py develop
opps-admin.py startproject PROJECT_NAME
cd PROJECT_NAME
```

```
python manage.py syncdb --noinput
python manage.py migrate
python manage.py collectstatic --noinput
python manage.py runserver
```

8.2.3 Installation

You can use pip to install Opps and requirements

```
pip install opps
```

or

```
git clone git@github.com:opps/opps.git
cd opps
python setup.py install
```

8.2.4 Start project

You can use *opps-admin.py* to start new project

```
opps-admin.py startproject PROJECT_NAME
```

8.3 Configuration

8.4 Opps Admin Command-line

opps-admin.py is Opps's command-line utility for administrative tasks.

8.4.1 Usage

command should be one of the commands listed in this document. *options*, which is optional, should be zero or more of the options available for the given command.

```
opps-admin.py <command> [options]
```

8.5 Getting runtime help

Run *opps-admin.py* help to display usage information and a list of the commands provided by each application.

8.6 startproject <projectname>

Creates a Opps project directory structure for the given project name in the current directory or the given destination. For example:

```
opps-admin.py startproject myproject
```

Opps will also accept URLs (http, https, ftp) to compressed archives with the project template files, downloading and extracting them on the fly.

8.7 Template Customization

8.7.1 Opps default

- Channel: templates/containers/list.html
- Content: templates/containers/detail.html

8.7.2 Details

- containers/<channel-slug>/<sub-channel-slug>/<container-slug>/detail.html
- containers/<channel-slug>/<sub-channel-slug>/<container-child-class>_detail.html
- containers/<channel-slug>/<sub-channel-slug>/detail.html
- containers/<channel-slug>/<container-child-class>_detail.html
- containers/<channel>/detail.html
- containers/detail.html

8.7.3 Channel conf

All channel configuration is one json file, a channel can have more than one layout, changed when editing channel (file name **templates/containers/<channel-name>/channel.json**):

```
{"layout": ["home_1", "home_2"]}
```

8.8 Content Architecture

8.8.1 The `Container` Model

The conventions of a Opps site is the modelo `opps.containers.models.Container`. Each `container` instance is stored in a hierarchical tree to form the content navigation, and an interface for managing the structure of the navigation tree is provided in the admin via `opps.containers.admin.ContainerAdmin`.

8.8.2 Creating Custom Content Type

In order to handle different content type that require more structured content than provided by the `Post`, `Album` or `Link` model, you can simply create your own models that inherit from `Container`. For example if we wanted to have content that were **music**, **author** and **studio**:

```
# -*- coding: utf-8 -*-
from django.db import models
from opps.containers.models import Container


class Musics(Container):
    music = models.CharField(_(u"Music"), max_length=140)
    author = models.CharField(_(u"Author"), max_length=140)
    studio = models.CharField(_(u"Studio"), max_length=200)

    class META:
        verbose_name = _(’Music’)
        verbose_name_plural = _(’Musics’)
```

Next you’ll need to register your model with Django’s admin to make it available as a content type. If your content type only exposes some new fields that you’d like to make editable in the admin, you can simply register your model using the `opps.containers.admin.ContainerAdmin` class:

```
# -*- coding: utf-8 -*-
from django.contrib import admin
from opps.containers.admin import ContainerAdmin
from .models import Musics

admin.site.register(Musics, ContainerAdmin)
```

8.9 API

8.9.1 Get all containers object

- /api/v1/container/?format=json

8.9.2 Get container object

- /api/v1/<child class>/<id>/?format=json
- /api/v1/post/<id>/?format=json
- /api/v1/album/<id>/?format=json
- /api/v1/link/<id>/?format=json

8.10 Opps Customization

8.10.1 Opps Editor

By default it uses Tinymce

To use aloha

```
OPPS_EDITOR = {
    'editor': 'aloha',
    'js': ("//cdn.aloha-editor.org/latest/lib/require.js",),
    "css": ("//cdn.aloha-editor.org/latest/css/aloha.css",)
}
```

To use redactor

Install redactor:

```
pip install django-wysiwyg-redactor
```

Add django redactor to your INSTALLED_APPS:

Add django redactor to your urls.py:

```
urlpatterns = patterns(
    '',
    url(r'^media/(?P<path>.*)$', 'django.views.static.serve',
    {'document_root': settings.MEDIA_ROOT, 'show_indexes': True}),
    url(r'^', include('opps.urls')),
    # django redactor
    url(r'^redactor/', include('redactor.urls')),
)
```

Set your own static paths:

```
OPPS_EDITOR = {
    'editor': 'redactor',
    'js': ('redactor/redactor.min.js',),
    "css": ('redactor/css/redactor.css',
            'redactor/css/django_admin.css')
}
```

8.11 Opps Application

- opps.polls
- opps.infographics
- opps.ganalytics
- opps.promos
- opps.timelinejs
- opps.registration
- more...

8.12 Packages

8.12.1 opps.api

8.12.2 opps.archives

8.12.3 opps.articles

8.12.4 opps.boxes

8.12.5 opps.channels

8.12.6 opps.containers

8.12.7 opps.contrib.feeds

8.12.8 opps.contrib.fileupload

8.12.9 opps.contrib.logging

Application to generate log project actions

Example

```
$.ajax({
    type: 'POST',
    url: '/api/v1/contrib/logging/',
    data: '{"application": "player", "action": "play"}',
    dataType: "application/json",
    processData:  false,
    contentType: "application/json"
});
```

8.12.10 opps.contrib.mobile

8.12.11 opps.contrib.multisite

8.12.12 opps.contrib.notifications

8.12.13 opps.core.tags

8.12.14 opps.core.templatetags

8.12.15 opps.db._redis

8.12.16 opps.db.backends.postgresql_psycopg2

8.12.17 opps.fields

8.12.18 opps.flatpages

8.12.19 opps.images

8.12.20 opps.search

8.12.21 opps.sitemaps

8.12.22 opps.views

8.13 Git Tips

While some of this information repeats documentation that is already available on GitHub Help, we thought it might be useful to collect and publish some of the most relevant information.

8.13.1 Forking Opps

Assuming you have already set up Git, the first step is to navigate to the Opps project page and select the Fork button at top-right.

Next, clone your fork (replace GH-USER with your GitHub username):

```
git clone https://github.com/GH-USER/opps.git
```

This adds a “remote” for your fork called “origin”. Add the canonical Opps project as an additional remote called “upstream”:

```
cd opps
git remote add upstream https://github.com/opps/opps.git
```

8.13.2 Making your changes

Create and switch to a new branch to house your feature or bugfix (replace *newfeaturebranch* with an appropriate name):

```
git fetch upstream
git checkout -b newfeaturebranch upstream/master
```

Once you are satisfied with your changes, run the tests and check coding standards:

```
make test
```

Once the tests all pass and you are comfortable that your code complies with the suggested coding standards, add and commit your changes:

```
git add changedfile1 changedfile2
git commit
```

Push your new branch, and the commit(s) within, to your fork:

```
git push origin newfeaturebranch
```

8.13.3 Issuing a pull request

Read the GitHub pull request documentation first. Then navigate to your Opps fork at *https://github.com/GH-USER/opps*, select the new branch from the drop-down, and select “Pull Request”. Enter a title and description for your pull request, review the proposed changes using the “Commits” and “Files Changed” tabs, and select “Send pull request”.

If someone reviews your contribution and ask you to make more changes, do so and then rebase to upstream master:

```
git checkout newfeaturebranch
git fetch upstream
git rebase -p upstream/master
```

When prompted, include a relevant commit message, describing all your changes. Finally, push your changes via:

```
git push --force origin newfeaturebranch
```

8.13.4 Squashing commits

If you are asked to squash your commits:

```
git checkout newfeaturebranch
git fetch upstream
git rebase upstream/master
git rebase -i
```

When prompted, mark your initial commit with pick, and all your follow-on commits with squash.

Then edit the commit message to make sense, taking out any extraneous information and succinctly describing your changes. Finally, push your changes via:

```
git push --force origin newfeaturebranch
```

8.13.5 Relevant Resources

- Git merge vs. rebase

8.14 Develop

8.14.1 South Migrations

Note: Remember this for every South migration created!

For every migration created, make the following modifications.

```
from django.contrib.auth import get_user_model
User = get_user_model()


class Migration(SchemaMigration):

    def forwards(self, orm):
        # reapeat the following for every create or alter table using "user" relation
        db.create_table('app.model', (
            ('user', self.gf('django...ForeignKey')(to=orm["%s.%s" % (User._meta.app_label, User._met

    models = {
        ...
        # this should replace "auth.user or accounts.customuser"
        "%s.%s" % (User._meta.app_label, User._meta.module_name): {
            'Meta': {'object_name': User.__name__},
        }

        # repeat the following for every freezed model
        "app.model": {
            'user': ('django.db.models.fields.related.ForeignKey', [], {'to': "orm['%s.%s']"% (User._
        }
    }
```

8.14.2 Haystack

Note: This is required in order to use the default Haystack setup

The default haystack search template is the same you find in haystack home page, the difference in opps is that for every indexed Model you should implement some fields, properties or methods.

Required for search

- title (The title used on search result)
- search_category (It can be used to group things on tabs or just to show the category)
- get_absolute_url (The url for search result)
- get_thumb (optional image path for search result)

Example

```
class MyModel(models.Model):
    title = models.CharField(max_length=255) # implemented as field
    slug = models.SlugField()
    image = models.FileField()
    # implemented as method
    def get_absolute_url(self):
        return "posts/{0}".format(self.slug)

    # implemented as method
    def get_thumb(self):
        return get_thumb_url_or_something(self.image)

    # implemented as property
    @property
    def search_category(self):
        return _(' Blog post')
```

With the above in your model, you can now create your search_indexes and template and choose to index those fields/properties/methods or just access directly on template. (see haystack docs)

Example of search template

```
{% load images_tags %}


<h2>Search</h2>

<form method="get" action=".">
    <table>
        <input type="search" id="q" name="q" placeholder="Search" value="{{ request.GET.q}}" requirec
        <tr>
            <td>&nbsp;</td>
            <td>
                <input type="submit" value="Search">
            </td>
        </tr>
    </table>

    {% if query %}
        <h3>Results</h3>

        {% for result in page.object_list %}
            <p>
              <small>{{ result.object.search_category }}</small><br>
              {% if result.object.get_thumb %}
              <a href="{{ result.object.get_absolute_url }}">
                  <img src="{% image_url result.object.get_thumb.archive.url width=100 height=100 %}"
              </a>
              {% endif %}

                <a href="{{ result.object.get_absolute_url }}">{{ result.object.title }}</a>
            </p>
        {% empty %}
            <p>No results found.</p>
        {% endfor %}

        {% if page.has_previous or page.has_next %}
            <div>
                {% if page.has_previous %}<a href="?q={{ query }}&amp;page={{ page.previous_page_numb
                |
                {% if page.has_next %}<a href="?q={{ query }}&amp;page={{ page.next_page_number }}">
            </div>
        {% endif %}
    {% else %}
        {# Show some example queries to run, maybe query syntax, something else? #}
    {% endif %}
</form>
```

8.15 Contributing

Contributions are very welcome. Specially roles. If you implement a role that you think others might be using, please contribute.

To contribute head to opps's github page, fork it and create a pull request.

8.15.1 Developing

We strive to keep the internal quality of opps to the best that we can; Therefore, it's very important to keep some things in mind when contributing with code for opps:

- Test everything you can, with automated tests. If possible, please develop code with TDD. If you're having a hard time building tests, don't hesitate to ask for help in the opps mailing list. We are happy to help you keep your code well-covered by tests;
- When writing actual code, follow the conventions in PEP 8 (except for maximum line length, which we don't follow because there are too many parts of the project that require large strings to be used);
- When writing docstrings, follow the conventions in PEP 257 (take a look at other docstrings in the project to get a feel of how we organize them);
  - Also, when writing docstrings for the API, provide examples of how that method or class works. Having a code example of a part of the API is really helpful for the user.

8.15.2 Developer Setup

Software required in OS:

- Redis Server
- SQLite
- Image Lib

Check out the code from the github project:

```
git clone git://github.com/opps/opps.git
cd opps
```

Create a virtualenv (the example here is with virtualenvwrapper) and install all development packages:

```
mkvirtualenv opps
pip install -r requirements_dev.txt
python setup.py develop
```

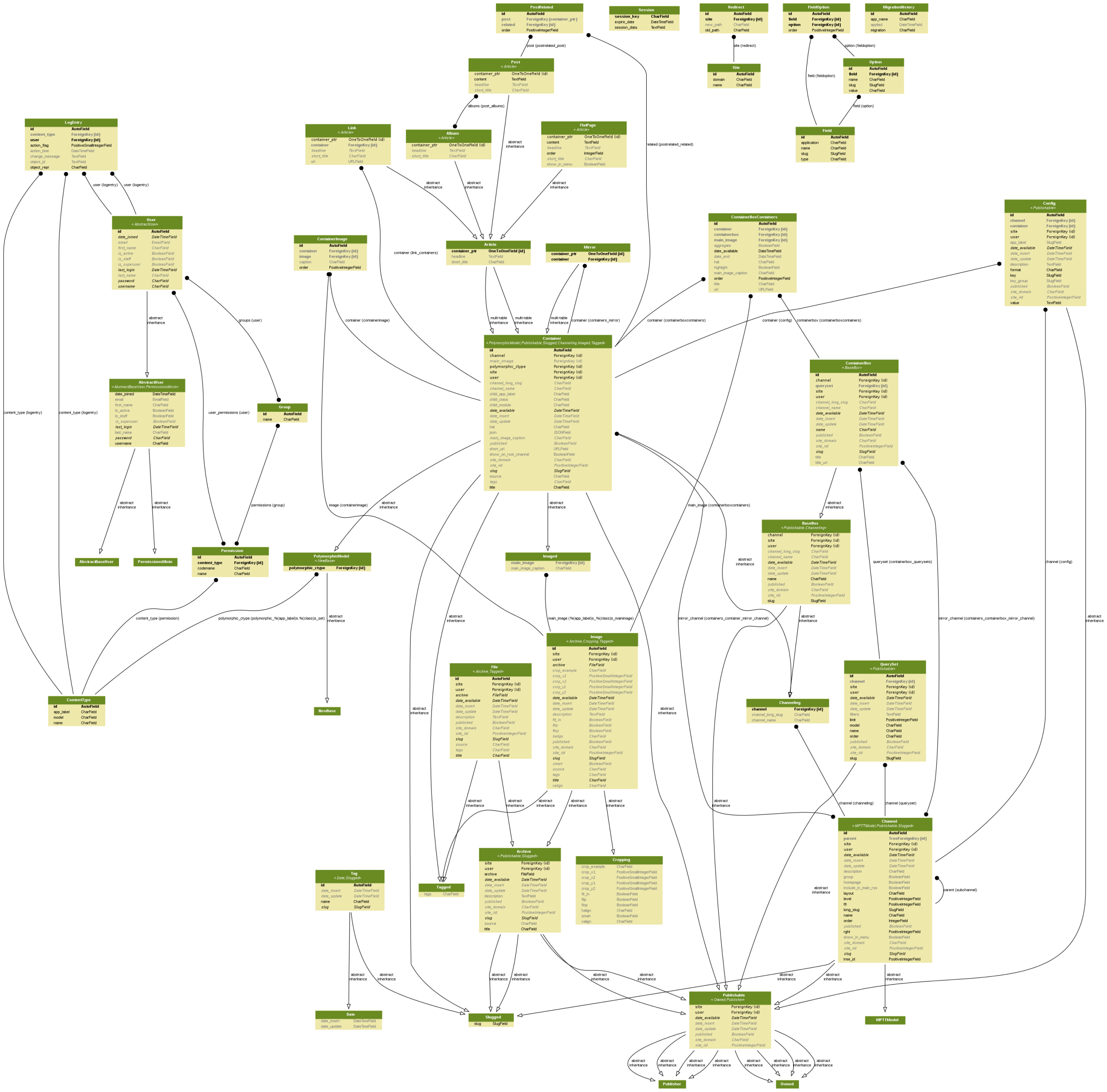
Here is how to run the test suite:

```
make test
```

Here is how to build the documentation:

```
cd docs
make html
```

8.15.3 Architecture



8.15.4 The team

The core team

The core team behind **opps** (in order of joining the project):

- Thiago Avelino (technical leader of this project)
- Bruno Rocha

- Patches and suggestions

Other contributors

Other non-core members, but equally important, equally rocking, equally ass-kicking contributors can be seen in this list: https://github.com/opps/opps/network/members

There are also some more contributors that haven't send code to the project, but who help in other ways, when and how they can. We're very happy to have you, guys!

Add member on TEAM

Clone repo oppsproject.org and add user on permissions.cfg in *[team:contributors]*

- Source (github)
- Mailing list

CHAPTER 9

Changelog

9.1 0.2.5

- Fixed fileupload error in fileupload-fp and add new file upload styles
- Fix home (channel) if not exist in mult-site child
- Periodic task to support recheck create mirror channel
- Fix *get_containerbox* templatetag recursion parameter issue

9.2 0.2.4

- Add endless pagination on requirements file, used on opps editor for locate image
- Add missing Notification model migration
- Update README file, add noimput in syncdb run
- Add new field recursive on Boxe (app) QuerySet, #320
- Fixed version dependency Pillow/Django-Filebrowser, #322
- Update MANIFEST.in and include all the search templates, #315

9.3 0.2.3

- Add custom sitemap view and OPPS_SITEMAP_LANGUAGE and OPPS_SITEMAP_PUBLICATION_NAME params
- Create template tag *exclude_queryset_by* on Containers application
- Create template tag *filter_queryset_by* on Containers application
- Change API Engine restframework to piston
-   - Easier to polymorphic work
-   - Old api removed
-   - Create *opps.api.ApiKeyAuthentication*
- Add Atom feed urls #119

- Fix image_obj template tag when sending Nonetype image
- Create Opps Vagrant box to help other contributors
- Added support to ajax requests with extends_parent variable in template context
- Write logging contrib application #275
- Fixed run tests on celery, because use Calling Tasks
- Update fixture example
- Fix test running on Django 1.5, 1.6 and 1.7 #145
-   – Change test folder, opps/<application>/tests to tests/<application>
-   – Used nose
- Create *OPPS_CORE_APPS*, recommend used on INSTALLED_APPS
- Get queryset (boxes) on get_containerbox (template tags containers), if exist queryset (on containerbox)
- Change ChannelListFilter. Now every parent channel will have an additional /* value on the lookups values
- Add context *breadcrumb* on get_context_data generic views (base)
- Add try_values and cache_obj template tags
- Change BaseBoxAdmin queryset permissions
- opps.contrib.mobile.middleware do not change `settings.TEMPLATE_DIRS` on the fly any more, it now use a thread-local variable
- Fix breadcrumb context variable
- Fix template tag *get_post_content*, change folder name articles to containers (Standard Opps 0.2.x)
- Fix filter_queryset_by and exclude_queryset_by when queryset is sliced
- Added extra_context to get_containerbox template tag
- Fix spaced and empty string tags creation
- Fix embedded album image order on Posts
- Containers in home page have direct url without channel, example: site.com/content_slug.html instead of site.com/home/content_slug.html
- Fix url pattern from flatpages, now accept slugs with dashs
- Add *get_custom_field_value* template tag
- Fix None hat field on Mirror creation
- Fix *main_image* caption population on Albums
- Add new *hat* field on Channel model
- Fix channel delete when it has some containers on it.
- Fix bug on mirror channel, if not used mirror channel resource, ref #310
- Fix TagList when home channel has a different layout. Issue #308
- Add Exclude field on QuerySet model of Boxes app. Issue #309

9.4 0.2.2

- Used argparse on opps-admin.py (bin) #82
- Fix test running on Django >= 1.6 #145
- More one channel per container (multi channel)
- Added raw_id_fields on ConfigAdmin
- fix bug, wrong crop params on image_obj templatetag, added lists of valid values
- Add field *title_url* on class model *ContainerBox*
- fix typo, settings_local.py with the wrong index for the database password 'PASS' is correct and 'PASSWORD'
- fix bug "List index out of range" in template tag get_containerbox_list
- Fix bug, mobile detect not bringing this path (url) #265
- Fix sitemaps and added a sitemaps index view
- Fix migration (auto user), ContainerBoxContainers add field highlight

9.5 0.2.1

- Add method `get_http_absolute_url` on channel model class
- Fix sitemap
- Remove contrib/db_backend , move to opps/db/backends #240
- Fix migrate run on postgresql - articles
- Add ChannelListFilter on HideContainerAdmin list_filter
- Add lazy translation on child_class list_display on HideContainerAdmin
- Add OPPS_CONTAINERS_BLACKLIST config on HideContainerAdmin
- Fix: image crop example
- Used get_descendants (mptt) on generic base view
- changing datetime.now to timezone.now on search index
- Fix unicode treatment JSONField rendering
- Write test on `opps.db._redis`
- Set dynamic db int, on db drive
- Fix: get_child recursivelly on template tag `get_container_by_channel`
- Changelog organize
- Fix docs organize
- Remove Opps theme docs, used default Read the Docs

9.6 0.2.0

- Content type (Container)
- Isoled boxes application
- ContainerBox, generic box (concept)
- Used Container in all application
- Archives, file manager
- Images used archives
- Used RST on README, pypi compatibility
- Add contrib pattern (like django)
- Upgrade haystack to 2.0 (stable)
- Opps Generic Views
- New view format, used to URLs pattern
- Add Grappelli dependence of the project
- Create Opps DB (NoSQL Database architecture)
- Add redis support (Opps BD)
- Contrib notification, central message exchange between container
- – websocket support
- – sse support
- – long pulling support
- Add field highlight on ContainerBox option
- Fix bug generic view list, get recursive channel list
- Dynamic fields on container, via JSONField
- – Text
- – Textarea
- – Checkbox
- – Radio
- Fix template tag `image_obj`
- Add optional container filtering by child_class in ListView
- fix flatpage url
- Adding .html in containers url

9.7 0.1.9

9.8 0.1.8

- Queryset cache on generic view

- Add image thumb on ArticleBox
- Send current site to template `{{ SITE }}`
- In /rss feed, filter channels by **published** and **include_in_main_rss**
- RSS Feed now renders in a template
- Flatpage is content type Article
- **Hotfix** fix *memory leak* (articles generic view)
- Chekc OPPS_PAGINATE_NOT_APP app not used PAGINATE_SUFFIX
- Used cache page

9.9 0.1.7

9.10 0.1.6

9.11 0.1.5

9.12 0.1.4

9.13 0.1.3

9.14 0.1.0

- Initial release

9.15 Indices

- *genindex*
- *search*