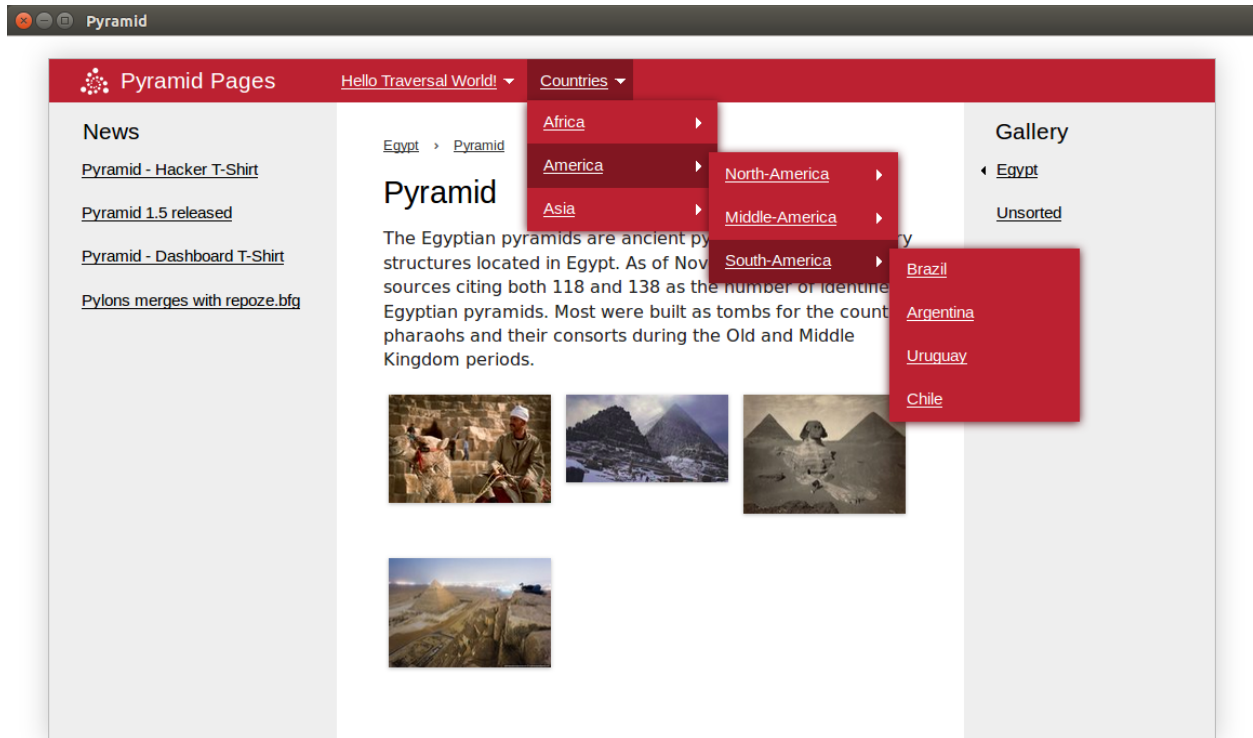

pyramid*pages*
Release

July 06, 2015

1	Installing	3
1.1	github	3
1.2	PyPi	3
1.3	source	3
2	Configuration	5
2.1	Custom model for tree pages	5
2.2	Configure <i>pyramid_pages</i>	5
2.3	Custom resource	6
2.4	Generate menu	6
3	API	11
3.1	common	11
3.2	models	11
3.3	routes	12
3.4	views	13
3.5	security	13
3.6	Module contents	13
4	Tutorials and Cookbook Recipes	15
4.1	Simple Web-site with tree pages	15
5	Support and Development	19
6	Indices and tables	21
	Python Module Index	23

pyramid_{pages} provides a collections of pages to your Pyramid application. This is very similar to **django.contrib.flatpages** but with a tree structure and traversal algorithm in URL dispath.



Installing

1.1 github

```
pip install git+http://github.com/ITCase/pyramid_pages.git
```

1.2 PyPi

```
pip install pyramid_pages
```

1.3 source

```
git clone git+http://github.com/ITCase/pyramid_pages.git  
python setup.py develop
```

Configuration

2.1 Custom model for tree pages

To build a tree, using the model from `sqlalchemy_mptt`.

Note:

Otherwise, it will look like flat pages.

Plans to add a recursive model with only *parent_id* field.

Create model of tree pages. For more detail see example `pyramid_pages_example`.

```
from pyramid_pages.models import FlatPageMixin, MpttPageMixin, RedirectMixin
from sqlalchemy_mptt import BaseNestedSets

...

class WebPage(Base, MpttPageMixin, RedirectMixin):
    __tablename__ = 'mptt_pages'

    id = Column('id', Integer, primary_key=True)

class NewsPage(Base, FlatPageMixin):
    __tablename__ = 'flat_news'

    id = Column('id', Integer, primary_key=True)
    date = Column(Date, default=func.now())
```

2.2 Configure *pyramid_pages*

Then add settings of *pyramid_pages*.

```
from youproject.models import MPTTPages, MPTTNews

...

settings['pyramid_pages.models'] = {
    '': WebPage,
    'pages': WebPage,  # available with prefix '/pages/'
```

```
'news': NewsPage
}

# pyramid_pages - put it after all routes
# and after pyramid_pages configuration.
config.include("pyramid_pages")
```

If you use version of pyramid $\geq 1.6a1$, there is a possibility put `config.include("pyramid_pages")` before pyramid_{pages} Configuration.

```
from youproject.models import MPTTPages, MPTTNews

...

config.include("pyramid_pages")
settings['pyramid_pages.models'] = {
    '': WebPage,
    'pages': WebPage, # available with prefix '/pages/'
    'news': NewsPage
}
```

2.3 Custom resource

Base resource for pages can be found in the module `pyramidpages.routes`.

Just inherit your resource from `PageResource`.

And add it to config.

```
settings['pyramid_pages.models'] = {
    '': WebPage,
    'pages': WebPage, # available with prefix '/pages/'
    'news': NewsPage,
    'gallery': GalleryResource
}
```

2.4 Generate menu

Make menu object and pass it in template context.

```
from pyramidpages.common import Menu

class Gallery(Base, MpttPageMixin):
    __tablename__ = 'mptt_gallery'

    menu_template = 'myproject/templates/my_custom_menu.mako'

    id = Column('id', Integer, primary_key=True)

page_menu = Menu(DBSession, WebPage).mptt
news_menu = Menu(DBSession, NewsPage).flat
gallery_menu = Menu(DBSession, Gallery).mptt
```

Just include menu template.

Listing 2.1: Base resource for pages.

```

1 class PageResource(object):
2
3     template = 'pyramid_pages/index.jinja2'
4     view = PageView
5     attr = 'page_with_redirect'
6
7     def __init__(self, node, prefix=None):
8         self.node = node
9         self.prefix = prefix
10
11     @property
12     def __name__(self):
13         if self.node and not self.node.slug == '/':
14             return self.node.slug
15         elif self.node and self.node.slug == '/':
16             return ''
17         elif self.prefix:
18             return self.prefix
19         return None
20
21     @property
22     def __parent__(self):
23         if hasattr(self.node, 'parent'):
24             return self.__class__(self.node.parent, self.prefix)
25         elif self.node:
26             return self.__class__(None, self.prefix)
27         return None
28
29     def __getitem__(self, name):
30         children = {str(child.slug or ''): self.__class__(child, self.prefix)
31                     for child in self.node.children}
32         return children[name]
33
34     def __resource_url__(self, request, info):
35         separator = '/' if self.prefix else ''
36         # XXX: I feel a dissonance here
37         info['virtual_path'] = re.sub('/+', '/', info['virtual_path'])
38         url = info['app_url'] + separator + info['virtual_path']
39         return url
40
41     def __repr__(self):
42         if not hasattr(self.node, 'name'):
43             return '{}'.format(self.node)
44         return "<{}>".format(self.node.name.encode('utf-8'))
45
46     def get_prefix(self, request, node=None):
47         if not node:
48             node = self.node
49         node = node.__class__
50         settings = request.registry.settings
51         models = settings[CONFIG_MODELS]
52         reversed_models = dict(zip(models.values(), models.keys()))
53         prefix = reversed_models.get(node, None)
54         if prefix:
55             return prefix
56
57         # search prefix for resource
58         for model in models.values():
59             if not hasattr(model, '__table__') and hasattr(model, 'model') \
60                 and model.model == node:
61                 return reversed_models.get(model, None)
62         return None

```

Listing 2.2: Custom resource for gallery.

```
1 class GalleryResource(PageResource):
2     model = Gallery
3     template = 'gallery/index.jinja2'
```

Listing 2.3: Model for GalleryResource.

```
1 class Gallery(BasePage, MpttPageMixin):
2     __tablename__ = 'mptt_gallery'
3
4     id = Column(Integer, ForeignKey('base_pages.id'), primary_key=True)
5
6     __mapper_args__ = {
7         'polymorphic_identity': 'gallery_page',
8     }
```

```
{% with menu=news_menu() %}
    {% include menu.template with context %}
{% endwith %}

{% with menu=page_menu(from_lvl=1, to_lvl=6, trees=(1, 2, 3)) %}
    {% include menu.template with context %}
{% endwith %}
```

Or write your own.

If you want show mptt menu with `to_lvl==max-2` or similar, just use `to_lvl=-2`.

```
{% with menu=page_menu(from_lvl=1, to_lvl=-2, trees=(1, 2, 3)) %}
    {% include menu.template with context %}
{% endwith %}
```

Listing 2.4: Model photo for Gallery.

```
1 class Photo(Base):
2     __tablename__ = 'photos'
3
4     id = Column('id', Integer, primary_key=True)
5     path = Column('path', Text)
6     gallery_id = Column(Integer, ForeignKey('mptt_gallery.id'))
7     gallery = relationship('Gallery', backref='photos')
```

Listing 2.5: Flat menu template.

```

1 <div class="main-menu-list">
2   {% for node in menu %}
3     <div class="main-menu-list__item
4       {% if node == context.node or node == page %}
5         main-menu-list__item_state_current
6       {% endif %}">
7       <a href="{{ request.resource_url(
8         context.__class__(node,
9         context.get_prefix(request, node=node))) }}"
10         class="main-menu-list__item-link">
11         {{ node.name }}
12       </a>
13     </div>
14   {% endfor %}
15 </div>

```

Listing 2.6: MPTT menu template.

```

1 <div class="main-menu-list">
2   {% for node, children in menu recursive %}
3     <div class="main-menu-list__item
4       {% if node == context.node or node == page %}
5         main-menu-list__item_state_current
6       {% endif %}
7       {% if children|length %}
8         main-menu-list__item_state_has-child
9       {% endif %}">
10      <a href="{{ request.resource_url(
11        context.__class__(node,
12        context.get_prefix(request, node=node))) }}"
13        class="main-menu-list__item-link">
14        {{ node.name }}
15      </a>
16      <div class="main-menu-list">{{ loop(children.items()) }}</div>
17    </div>
18  {% endfor %}
19 </div>

```


3.1 common

Menu generator

```
class pyramid_pages.common.Menu(session, model)
    Bases: object

    flat()

    mptt(from_lvl=1, **kwargs)

class pyramid_pages.common.PageMenu(items, template)
    Bases: object

pyramid_pages.common.recursive_node_to_dict(node, menu, json=None, json_fields=None)

pyramid_pages.common.sort_by_left(tree)
    Sort dict by left and tree_id field
```

3.2 models

Models for page.

```
class pyramid_pages.models.BaseSacrudMpttPage
    Bases: pyramid_pages.models.SeoMixin, pyramid_pages.models.RedirectMixin,
           pyramid_pages.models.MpttPageMixin

    verbose_name = 'MPTT pages'

class pyramid_pages.models.FlatPageMixin
    Bases: pyramid_pages.models.PageMixin

    get_menu()

    menu_template = 'pyramid_pages/menu/flat.jinja2'

class pyramid_pages.models.MpttPageMixin
    Bases: sqlalchemy_mptt.mixins.BaseNestedSets, pyramid_pages.models.PageMixin

    get_menu(**kwargs)

    menu_template = 'pyramid_pages/menu/mptt.jinja2'
```

```
class pyramid_pages.models.PageMixin
    Bases: object

    description = Column(None, UnicodeText(), table=None)
    in_menu = Column(None, Boolean(), table=None)
    name = Column(None, String(), table=None, nullable=False)
    slug = Column(None, SlugType(), table=None, nullable=False)
    visible = Column(None, Boolean(), table=None)

class pyramid_pages.models.RecursionPageMixin
    Bases: pyramid_pages.models.PageMixin

    model with single parent_id field

class pyramid_pages.models.RedirectMixin
    Bases: object

    redirect_type = Column(None, ChoiceType(), table=None)
    redirect_url = Column(None, String(), table=None)

class pyramid_pages.models.SeoMixin
    Bases: object

    seo_description = Column(None, String(), table=None)
    seo_keywords = Column(None, String(), table=None)
    seo_metatags = Column(None, UnicodeText(), table=None)
    seo_title = Column(None, String(), table=None)
```

3.3 routes

Routes for `sacrud_pages`

```
class pyramid_pages.routes.PageResource (node, prefix=None)
    Bases: object

    attr = 'page_with_redirect'
    get_prefix (request, node=None)
    template = 'pyramid_pages/index.jinja2'
    view
        alias of PageView

pyramid_pages.routes.home_page_factory (request)
pyramid_pages.routes.includeme (config)
pyramid_pages.routes.page_factory (request)
pyramid_pages.routes.register (*args)
```


3.4 views

Views for pages

```
class pyramid_pages.views.PageView(context, request)
    Bases: object
    page_with_redirect()
```

3.5 security

3.6 Module contents

```
pyramid_pages.includeme(config)
```

Tutorials and Cookbook Recipes

4.1 Simple Web-site with tree pages

This is an example of using `pyramid_pages` in just one file (`pyramid_pages_example.py`). Full code you can see [there](#).

```
1  #!/usr/bin/env python
2  # -*- coding: utf-8 -*-
3  # vim:fenc=utf-8
4  #
5  # Copyright © 2014 uralbash <root@uralbash.ru>
6  #
7  # Distributed under terms of the MIT license.
8
9  """
10 Main for example
11 """
12 import json
13 import os
14
15 import transaction
16 from pyramid.config import Configurator
17 from pyramid.events import BeforeRender
18 from pyramid.location import lineage
19 from pyramid.session import SignedCookieSessionFactory
20 from sqlalchemy import (Column, Date, ForeignKey, Integer, String, Text,
21                        engine_from_config)
22 from sqlalchemy.ext.declarative import declarative_base
23 from sqlalchemy.orm import relationship, scoped_session, sessionmaker
24 from sqlalchemy.sql import func
25 from zope.sqlalchemy import ZopeTransactionExtension
26
27 from pyramid_pages.common import Menu
28 from pyramid_pages.models import FlatPageMixin, MpttPageMixin, RedirectMixin
29 from pyramid_pages.routes import PageResource
30 from sqlalchemy_mptt import mptt_sessionmaker
31
32 Base = declarative_base()
33 DBSession = scoped_session(
34     mptt_sessionmaker(
35         sessionmaker(extension=ZopeTransactionExtension())
36     )
37 )
38
```

```
39 CONFIG_SQLALCHEMY_URL = 'sqlalchemy.url'
40 CONFIG_PYRAMID_PAGES_MODELS = 'pyramid_pages.models'
41 CONFIG_PYRAMID_PAGES_DBSESSION = 'pyramid_pages.dbsession'
42
43
44 class BasePage(Base, RedirectMixin):
45     __tablename__ = 'base_pages'
46     id = Column(Integer, primary_key=True)
47     page_type = Column(String(50))
48
49     __mapper_args__ = {
50         'polymorphic_identity': 'base_page',
51         'polymorphic_on': page_type
52     }
53
54     @classmethod
55     def get_pk_name(cls):
56         return 'id'
57
58     @classmethod
59     def get_pk_with_class_name(cls):
60         return 'BasePage.id'
61
62
63 class WebPage(BasePage, MpttPageMixin):
64     __tablename__ = 'mptt_pages'
65
66     id = Column(Integer, ForeignKey('base_pages.id'), primary_key=True)
67
68     __mapper_args__ = {
69         'polymorphic_identity': 'web_page',
70     }
71
72
73 class NewsPage(BasePage, FlatPageMixin):
74     __tablename__ = 'flat_news'
75
76     id = Column(Integer, ForeignKey('base_pages.id'), primary_key=True)
77     date = Column(Date, default=func.now())
78
79     __mapper_args__ = {
80         'polymorphic_identity': 'news_page',
81     }
82
83
84 class Gallery(BasePage, MpttPageMixin):
85     __tablename__ = 'mptt_gallery'
86
87     id = Column(Integer, ForeignKey('base_pages.id'), primary_key=True)
88
89     __mapper_args__ = {
90         'polymorphic_identity': 'gallery_page',
91     }
92
93
94 class Photo(Base):
95     __tablename__ = 'photos'
96
```

```

97     id = Column('id', Integer, primary_key=True)
98     path = Column('path', Text)
99     gallery_id = Column(Integer, ForeignKey('mptt_gallery.id'))
100    gallery = relationship('Gallery', backref='photos')
101
102
103    class GalleryResource(PageResource):
104        model = Gallery
105        template = 'gallery/index.jinja2'
106
107
108    class NewsResource(PageResource):
109        model = NewsPage
110        template = 'news/index.jinja2'
111
112
113    class Fixtures(object):
114
115        def __init__(self, session):
116            self.session = session
117
118        def add(self, model, fixtures):
119            here = os.path.dirname(os.path.realpath(__file__))
120            file = open(os.path.join(here, fixtures))
121            fixtures = json.loads(file.read())
122            for fixture in fixtures:
123                self.session.add(model(**fixture))
124                self.session.flush()
125            transaction.commit()
126
127
128    def add_globals(event):
129        event['lineage'] = lineage
130        event['page_menu'] = Menu(DBSession, WebPage).mptt
131        event['news_menu'] = Menu(DBSession, NewsPage).flat
132        event['gallery_menu'] = Menu(DBSession, Gallery).mptt
133
134
135    models = {
136        '': WebPage,
137        'pages': WebPage,
138        'news': NewsResource,
139        'gallery': GalleryResource,
140    }
141
142
143    def main(global_settings, **settings):
144        config = Configurator(
145            settings=settings,
146            session_factory=SignedCookieSessionFactory('itsaseekreet')
147        )
148        config.include('pyramid_jinja2')
149        config.add_jinja2_search_path('pyramid_pages_example:templates')
150        config.add_static_view('pyramid_pages_example_static',
151                               'static')
152
153        # Database
154        settings = config.get_settings()

```

```
155     settings[CONFIG_SQLALCHEMY_URL] =\  
156         settings.get (CONFIG_SQLALCHEMY_URL,  
157             'sqlite:///example.sqlite')  
158     engine = engine_from_config(settings)  
159     DBSession.configure(bind=engine)  
160     Base.metadata.drop_all(engine)  
161     Base.metadata.create_all(engine)  
162     fixture = Fixtures(DBSession)  
163     fixture.add(WebPage, 'fixtures/pages.json')  
164     fixture.add(WebPage, 'fixtures/country.json')  
165     fixture.add(NewsPage, 'fixtures/news.json')  
166     fixture.add(Gallery, 'fixtures/gallery.json')  
167     fixture.add(Photo, 'fixtures/photos.json')  
168  
169     # pyramid_pages  
170     settings[CONFIG_PYRAMID_PAGES_DBSESSION] =\  
171         settings.get (CONFIG_PYRAMID_PAGES_DBSESSION,  
172             DBSession)  
173     settings[CONFIG_PYRAMID_PAGES_MODELS] =\  
174         settings.get (CONFIG_PYRAMID_PAGES_MODELS, models)  
175     config.include("pyramid_pages")  
176     config.add_subscriber(add_globals, BeforeRender)  
177     return config.make_wsgi_app()  
178  
179 if __name__ == '__main__':  
180     settings = {}  
181     app = main({}, **settings)  
182  
183     from wsgiref.simple_server import make_server  
184     httpd = make_server('0.0.0.0', 6543, app)  
185     httpd.serve_forever()
```

Support and Development

To report bugs, use the [issue tracker](#) or [waffle board](#).

We welcome any contribution: suggestions, ideas, commits with new features, bug fixes, refactoring, docs, tests, translations etc

If you have question, contact me sacrud@uralbash.ru or IRC channel #sacrud

Indices and tables

- `genindex`
- `modindex`
- `search`

p

- `pyramid_pages`, [13](#)
- `pyramid_pages.common`, [11](#)
- `pyramid_pages.models`, [11](#)
- `pyramid_pages.routes`, [12](#)
- `pyramid_pages.security`, [13](#)
- `pyramid_pages.views`, [13](#)

A

attr (pyramid_pages.routes.PageResource attribute), 12

B

BaseSacrudMpttPage (class in pyramid_pages.models), 11

D

description (pyramid_pages.models.PageMixin attribute), 12

F

flat() (pyramid_pages.common.Menu method), 11
FlatPageMixin (class in pyramid_pages.models), 11

G

get_menu() (pyramid_pages.models.FlatPageMixin method), 11
get_menu() (pyramid_pages.models.MpttPageMixin method), 11
get_prefix() (pyramid_pages.routes.PageResource method), 12

H

home_page_factory() (in module pyramid_pages.routes), 12

I

in_menu (pyramid_pages.models.PageMixin attribute), 12
includeme() (in module pyramid_pages), 13
includeme() (in module pyramid_pages.routes), 12

M

Menu (class in pyramid_pages.common), 11
menu_template (pyramid_pages.models.FlatPageMixin attribute), 11
menu_template (pyramid_pages.models.MpttPageMixin attribute), 11
mptt() (pyramid_pages.common.Menu method), 11

MpttPageMixin (class in pyramid_pages.models), 11

N

name (pyramid_pages.models.PageMixin attribute), 12

P

page_factory() (in module pyramid_pages.routes), 12
page_with_redirect() (pyramid_pages.views.PageView method), 13
PageMenu (class in pyramid_pages.common), 11
PageMixin (class in pyramid_pages.models), 11
PageResource (class in pyramid_pages.routes), 12
PageView (class in pyramid_pages.views), 13
pyramid_pages (module), 13
pyramid_pages.common (module), 11
pyramid_pages.models (module), 11
pyramid_pages.routes (module), 12
pyramid_pages.security (module), 13
pyramid_pages.views (module), 13

R

RecursionPageMixin (class in pyramid_pages.models), 12
recursive_node_to_dict() (in module pyramid_pages.common), 11
redirect_type (pyramid_pages.models.RedirectMixin attribute), 12
redirect_url (pyramid_pages.models.RedirectMixin attribute), 12
RedirectMixin (class in pyramid_pages.models), 12
register() (in module pyramid_pages.routes), 12

S

seo_description (pyramid_pages.models.SeoMixin attribute), 12
seo_keywords (pyramid_pages.models.SeoMixin attribute), 12
seo_metatags (pyramid_pages.models.SeoMixin attribute), 12
seo_title (pyramid_pages.models.SeoMixin attribute), 12

SeoMixin (class in pyramid_{pages}.models), [12](#)
slug (pyramid_{pages}.models.PageMixin attribute), [12](#)
sort_by_left() (in module pyramid_{pages}.common), [11](#)

T

template (pyramid_{pages}.routes.PageResource attribute),
[12](#)

V

verbose_name (pyramid_{pages}.models.BaseSacrudMpttPage
attribute), [11](#)
view (pyramid_{pages}.routes.PageResource attribute), [12](#)
visible (pyramid_{pages}.models.PageMixin attribute), [12](#)