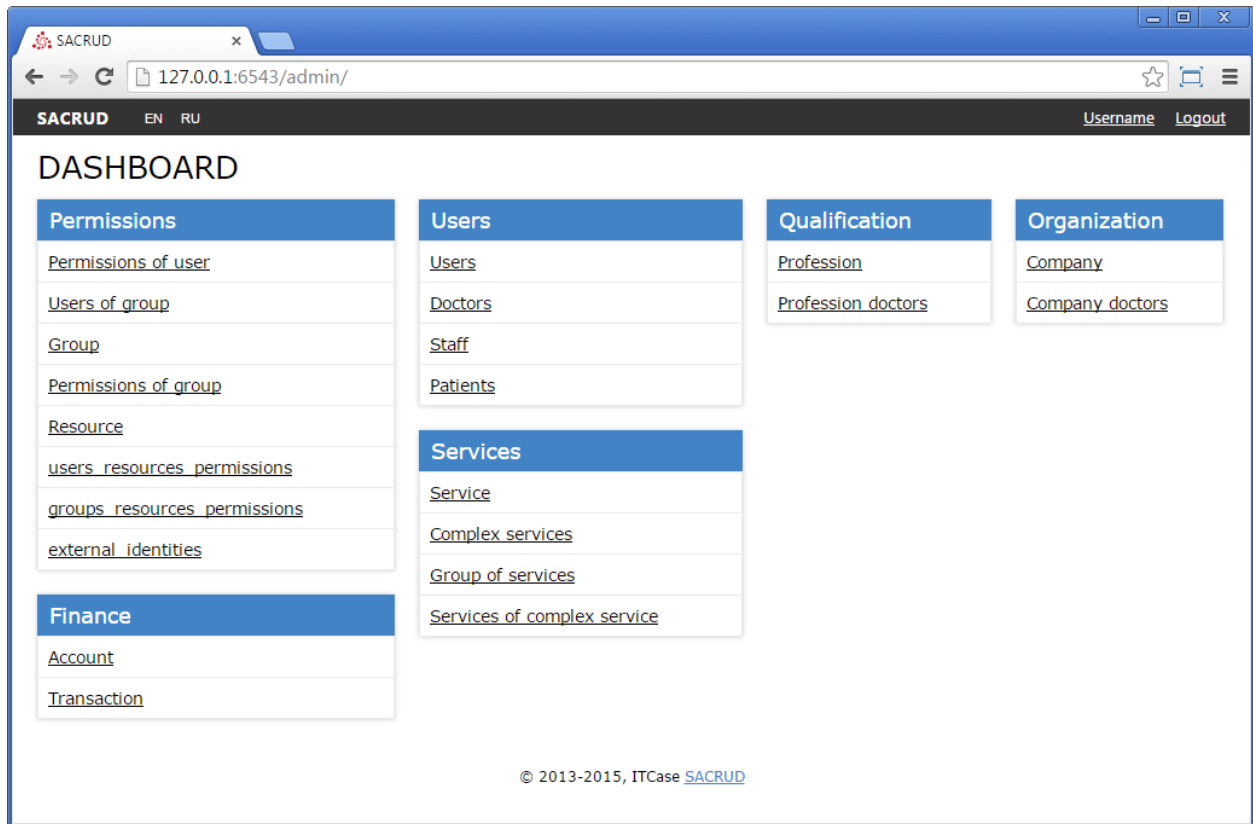

pyramid_s*acrud*
Release

July 06, 2015

1	Usage	3
1.1	Installing	3
1.2	Application Configuration for Pyramid	3
1.3	Localization	11
1.4	Permissions	15
1.5	API	15
2	Contribute	21
2.1	Backend contrubute	21
2.2	Frontend contribute	21
2.3	Localization contribute	24
3	Support and Development	27
4	License	29
5	Indices and tables	31
	Python Module Index	33



pyramid_sacrud - Pyramid CRUD interface based on [sacrud](#) and SQLAlchemy.

pyramid_sacrud will solve your problem of CRUD WEB interface for Pyramid. Unlike classical CRUD interface, *pyramid_sacrud* allows override and flexibly customize interface. (that is closer to *django.contrib.admin*)

Look how easy it is to use with Pyramid:

```
from .models import (Model1, Model2, Model3,)
# add sacrud and project models
config.include('pyramid_sacrud')
settings = config.registry.settings
settings['pyramid_sacrud.models'] = (('Group1', [Model1, Model2]),
                                     ('Group2', [Model3]))
```

go to <http://localhost:6543/sacrud/>

Example can be found here https://github.com/ITCase/pyramid_sacrud/tree/master/example

Usage

1.1 Installing

1.1.1 GitHub

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

1.1.2 PyPi

```
pip install pyramid_sacrud
```

1.1.3 Source

```
git clone git+http://github.com/ITCase/pyramid_sacrud.git
python setup.py install
```

1.1.4 Contribute

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

1.2 Application Configuration for Pyramid

1.2.1 Initialize

pyramid_sacrud use **Jinja2** template renderer

```
1 from .models import (Model1, Model2, Model3,)
2 # add pyramid_sacrud and project models
3 config.include('pyramid_sacrud')
4 settings = config.registry.settings
5 settings['pyramid_sacrud.models'] = (
6     ('Group1', [Model1, Model2]),
```

```
7 ('Group2', [Model13])
8 )
```

check it there <http://localhost:6543/sacrud/>

Set another prefix

```
config.include('pyramid_sacrud', route_prefix='admin')
```

now it there <http://localhost:6543/admin/>

1.2.2 Configure models

Model verbose name

```
1 class User(Base):
2
3     __tablename__ = 'user'
4
5     id = Column(Integer, primary_key=True)
6     name = Column(String)
7
8     def __init__(self, name):
9         self.name = name
10
11     # SACRUD
12     verbose_name = 'My user model'
```

Instead “user”, it will display “My user model”



Column verbose name

```
1 class User(Base):
2
3     __tablename__ = 'user'
4
5     id = Column(Integer, primary_key=True)
6     name = Column(String,
7                     info={"colanderalchemy": {'title': u'name of user'}})
8
```

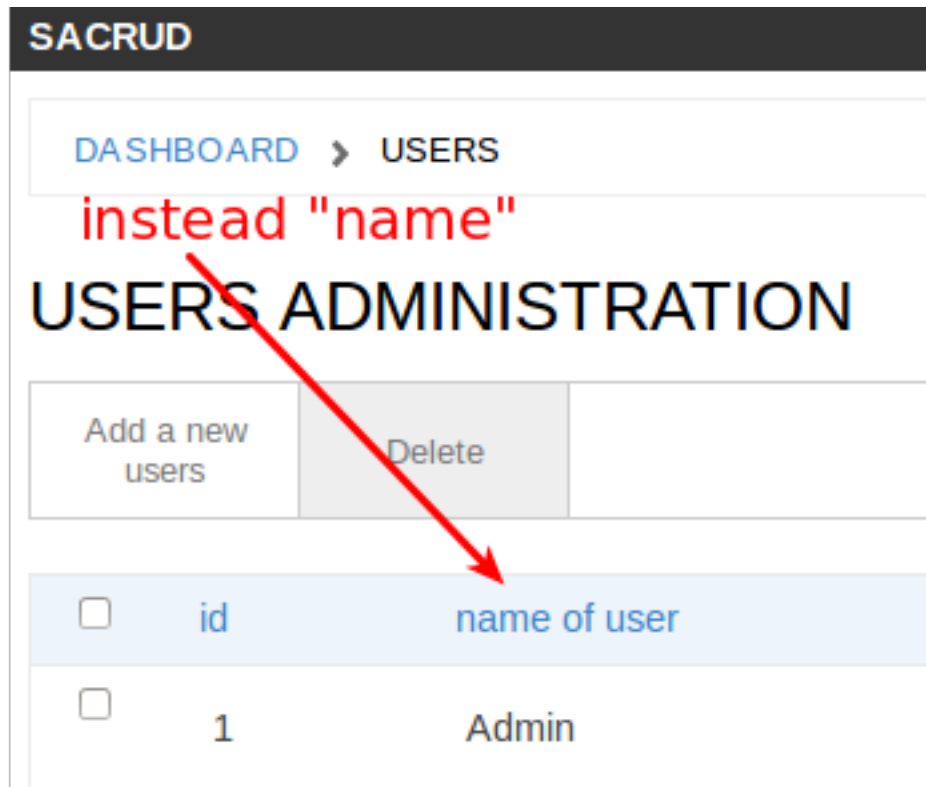


```

9  def __init__(self, name):
10     self.name = name

```

Instead “name”, it will display “name of user”



Description for column

```

1  class User(Base):
2
3      __tablename__ = 'user'
4
5      id = Column(Integer, primary_key=True)
6      name = Column(String,
7                    info={"colanderalchemy": {'title': u'name of user',
8                                              'description': 'put username there'}})
9
10     def __init__(self, name):
11         self.name = name

```

Adds a description below

SACRUD

DASHBOARD > USERS > ID=1

ADD A NEW USERS

Save	Delete	Cancel	
------	--------	--------	--

NAME SPACE

name

Admin

put there name

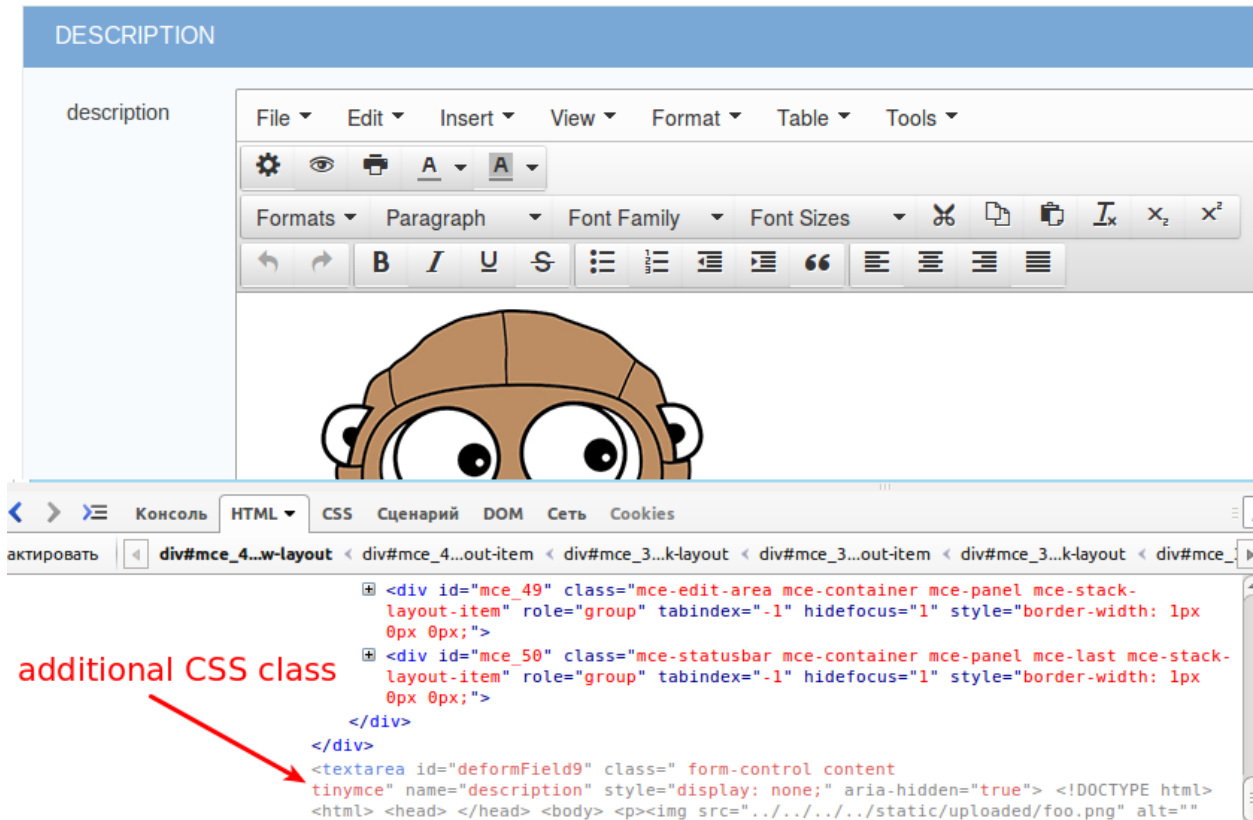
Add css class for column

```

1  tiny_mce_widget = deform.widget.TextAreaWidget(css_class='tiny_mce_content')
2
3
4  class TestCustomizing(Base):
5      __tablename__ = "test_customizing"
6
7      id = Column(Integer, primary_key=True)
8      description = Column(Text, info={'colanderalchemy': {'widget': tiny_mce_widget}})

```

Adds css class for column



Configure displayed fields in grid

```
1 class TestCustomizing(Base):
2     __tablename__ = "test_customizing"
3
4     id = Column(Integer, primary_key=True)
5     name = Column(String)
6     date = Column(Date)
7     name_ru = Column(String)
8     name_fr = Column(String)
9     name_bg = Column(String)
10    name_cze = Column(String)
11    description = Column(Text)
12    description2 = Column(Text)
13
14    sacrud_list_col = [name, name_ru, name_cze]
```

Use `sacrud_list_col` attribute of Model. It shows only `name`, `name_ru` and `name_cze` columns in grid.

<input type="checkbox"/>	name	Название	název
<input type="checkbox"/>	8E73A0		
<input type="checkbox"/>	2CAE6D		
<input type="checkbox"/>	1BB3D5		

Configure displayed columns for detailed object

```
1 class TestCustomizing(Base):
2     __tablename__ = "test_customizing"
3
4     id = Column(Integer, primary_key=True)
5     name = Column(String)
6     date = Column(Date)
7     name_ru = Column(String)
8     name_fr = Column(String)
9     name_bg = Column(String)
10    name_cze = Column(String)
11    description = Column(Text)
12    description2 = Column(Text)
13
14    sacrud_detail_col = [
15        ('name space', [name, name_ru, name_bg, name_fr, name_cze]),
16        ('description', [description, date, in_menu, visible, in_banner, description2])
17    ]
```

Use `sacrud_detail_col` attribute of Model. It agregate and composite columns in detail view.

NAME SPACE

name

8E73A0

put there name

i18n names

	Название
	Име
	nom
	název

DESCRIPTION


description

File Edit Insert View Format Table Tools

⚙️ 👁️ 🖨️ A A

Formats Paragraph Font Family Font Sizes ✂️ 📄

↶ ↷ B I U S ☰ ☷ ☹ ☹ ☹ ☹ ☹ ☹ ☹ ☹



1.2.3 Models attributes as property

Use `sacrud.common.TableProperty` decorator.

1.2.4 Composite fields and column as custom function

```
class WidgetRowLambda(object):
    name = "Name"
    info = {
        "sacrud_list_content": lambda x: ' '.join([x.surname, x.name, x.middlename])
    }
```

1.2. Application Configuration for Pyramid

9

Column as lambda function of row

```

1 class TestCustomizing(Base):
2     __tablename__ = "test_customizing"
3
4     id = Column(Integer, primary_key=True)
5     name = Column(String, info={"description": "put there name"})
6     surname = Column(String, info={"description": "put there name"})
7     middlename = Column(String, info={"description": "put there name"})
8     date = Column(Date, info={"verbose_name": 'date JQuery-ui'})
9     name_ru = Column(String, info={"verbose_name": u'', })
10    name_fr = Column(String, info={"verbose_name": u'nom', })
11    name_bg = Column(String, info={"verbose_name": u'', })
12    name_cze = Column(String, info={"verbose_name": u'název', })
13
14    sacrud_list_col = [
15        WidgetRowLambda(),
16        name_ru, name_cze]

```

<input type="checkbox"/>	user_name	email	Name
<input checked="" type="checkbox"/>	Edit MakarovES	foo123aaa	Ivanov Sergey Valerevich

Column as relationship

```

1 class User(Base):
2     __tablename__ = 'users'
3     verbose_name = _('Users')
4
5     id = Column(Integer, primary_key=True, autoincrement=True)
6     name = Column(Unicode, nullable=False)
7
8
9 class Company2User(Base):
10    __tablename__ = 'm2m_company2user'
11    verbose_name = _('Company of user')
12
13    user_id = Column(Integer, ForeignKey('user.id'), primary_key=True)
14    user = relationship("User",
15        info={'colanderalchemy': {'title': _("Users")}}
16    )
17    company_id = Column(Integer, ForeignKey('company.id'), primary_key=True)
18    company = relationship("Company",
19        info={'colanderalchemy': {'title': _("Affiliate company")}}
20    )
21
22
23 class Company(Base):
24     __tablename__ = 'company'
25     verbose_name = _('Company')
26
27     id = Column(Integer, primary_key=True, autoincrement=True)
28     name = Column(Unicode)
29     company_id = Column(Integer, ForeignKey('company.id'), nullable=True,
30        info={'verbose_name': _('Main company')})

```

```

31 company = relationship('Company')
32 users = relationship('User', secondary='m2m_company2user',
33                      backref='company')
34
35 # SACRUD
36 @TableProperty
37 def sacrud_detail_col(cls):
38     return [(' ', [Company.name, cls.c.company_id,
39                  Company.users, Company.company])]
40
41
42 def __repr__(self):
43     return self.name

```

ManyToMany

```

1 users = relationship('User', secondary='m2m_company2user',
2                      backref='company')
3
4 # SACRUD
5 @TableProperty
6 def sacrud_detail_col(cls):
7     return [(' ', [Company.name, cls.c.company_id, Company.users])]

```

1.2.5 Template redefinition

Just create template file in your project templates/sacrud directory

```

myapp/
-- templates
  -- sacrud
    -- home.jinja2 <-- custom template for pyramid_sacrud home page

```

You can also change the template for just one model or your own for each model.

```

1 class Tree(Base):
2     __tablename__ = 'tree'
3
4     id = Column(Integer, primary_key=True)
5     name = Column(Unicode)
6     parent_id = Column(Integer, ForeignKey('tree.id'))
7
8     sacrud_list_template = 'myapp:/templates/my_custom_list_with_jqtree.jinja2'
9     sacrud_edit_template = 'myapp:/templates/my_custom_edit.mako'

```

1.3 Localization

1.3.1 Translate widget name

To translate widgets on main page, you can use standart translate method for Pyramid.

```

from pyramid.i18n import TranslationStringFactory

_ = TranslationStringFactory('myapplication')
settings['pyramid_sacrud.models'] = (

```

SACRUD

DASHBOARD

Permissions
<u>Permissions of user</u>
<u>Users of group</u>
<u>Group</u>
<u>Permissions of group</u>
<u>Resource</u>

Fig. 1.1: http://localhost:6543/sacrud?_LOCALE_=en

SACRUD

ПАНЕЛЬ УПРАВЛЕНИЯ

Права
<u>Права</u>
<u>Пользователи группы</u>
<u>Группы</u>
<u>Права группы</u>
<u>Ресурсы</u>

Fig. 1.2: http://localhost:6543/sacrud?_LOCALE_=ru


```

    (_('Permissions'), (
        UserPermission,
        UserGroup,
        Group,
        GroupPermission,
        Resource,
        UserResourcePermission,
        GroupResourcePermission,
        ExternalIdentity,
    ))
),
(_('Users'), (User, Staff))
)

```

For more information see [Internationalization and Localization](#)

1.3.2 Translate model name

SACRUD

DASHBOARD
PERMISSIONS OF USER
 PERM_NAME:

PERMISSIONS OF USER

Save
 Delete
 Cancel

perm_name

user_id

Save
 Delete
 Cancel

http://localhost:6543/sacrud/users_permissions/update/perm_name/pyramid_sacrud_home/user_id/1?_LOCALE_=en
http://localhost:6543/sacrud/users_permissions/update/perm_name/pyramid_sacrud_home/user_id/1?_LOCALE_=ru

SACRUD

[ПАНЕЛЬ УПРАВЛЕНИЯ](#) [ПРАВА](#) PERM_NAME=PYRAM

ПРАВА

Сохранить

Удалить

Отмена

perm_name

pyramid_sacrud_home

user_id

Foo Baz Bar ▴ ▾

Сохранить

Удалить

Отмена

```
class GroupPermission(GroupPermissionMixin, Base):
    verbose_name = _('Permissions of group')
```

1.3.3 Translate group in edit/create form

```
from sacrud.common import TableProperty

class User(UserMixin, Base):
    verbose_name = _('Users')

    id = Column(Integer, primary_key=True, autoincrement=True)
    name = Column(Unicode, nullable=False)
    middlename = Column(Unicode, nullable=False)
    surname = Column(Unicode, nullable=False)

    def __repr__(self):
        return self.name + ' ' + self.middlename + ' ' + self.surname

    # SACRUD
    @TableProperty
    def sacrud_detail_col(cls):
        col = cls.columns
        return [('_', [col.user_name, col.email, col.user_password]),
                (_('personal data'), [col.name, col.middlename, col.surname])]
```

1.4 Permissions

If you use *RootFactory* for authorization, set next permissions for you aessed to pyramid_sacrud user.

pyramid_sacrud.security

```
PYRAMID_SACRUD_HOME = 'pyramid_sacrud_home'
PYRAMID_SACRUD_LIST = 'pyramid_sacrud_list'
PYRAMID_SACRUD_UPDATE = 'pyramid_sacrud_update'
PYRAMID_SACRUD_CREATE = 'pyramid_sacrud_create'
PYRAMID_SACRUD_DELETE = 'pyramid_sacrud_delete'
```

1.5 API

1.5.1 Main module

Includeme of SACRUD

`pyramid_sacrud.includeme` (*config*)

1.5.2 Breadcrumbs

Breadcrumbs for sacrud pyramid extension.

`pyramid_sacrud.breadcrumbs.breadcrumbs` (*name, verbose, view, id=None*)

SACRUD

[DASHBOARD](#) [USERS](#) ID=1

USERS

Save

Delete

Cancel

user_name	<input type="text" value="admin"/>
email*	<input type="text" value="foo@bar.baz"/>
user_password	<input type="text" value="123"/>

Personal Data

name*	<input type="text" value="Foo"/>
middlename*	<input type="text" value="Baz"/>
surname*	<input type="text" value="Bar"/>

Save

Delete

Cancel

Fig. 1.3: http://localhost:6543/admin/users/update/id/1?_LOCALE_=en

SACRUD

[ПАНЕЛЬ УПРАВЛЕНИЯ](#) [ПОЛЬЗОВАТЕЛИ](#) ID=1

ПОЛЬЗОВАТЕЛИ

Сохранить

Удалить

Отмена

user_name

admin

email*

foo@bar.baz

user_password

123

group "personal data"

Персональные Данные

name*

Foo

middlename*

Baz

surname*

Bar

Сохранить

Удалить

Отмена

Fig. 1.4: http://localhost:6543/admin/users/update/id/1?_LOCALE_=ru

```
pyramid_sacrud.breadcrumbs.get_crumb(name, visible, view, params)
```

1.5.3 Common

Any helpers for Pyramid

```
pyramid_sacrud.common.get_models_from_settings(settings)
```

```
pyramid_sacrud.common.get_settings_param(settings, name)
```

```
pyramid_sacrud.common.get_table(tname, request)
```

Return table by table name from pyramid_sacrud.models in settings.

```
pyramid_sacrud.common.get_table_verbose_name(table)
```

```
pyramid_sacrud.common.import_from_string(path)
```

```
pyramid_sacrud.common.pkg_prefix(config)
```

Function for return pkg prefix.

```
>>> from pyramid.config import Configurator
>>> settings = {'foo': 'foo', 'bar': 'bar'}
```

```
# Create config >>> config = Configurator(settings=settings)
```

```
# w/o route_prefix >>> pkg_prefix(config) '/sacrud/'
```

```
# with route_prefix >>> config.route_prefix = "/admin" >>> pkg_prefix(config) "
```

```
pyramid_sacrud.common.preprocessing_value(value)
```

```
pyramid_sacrud.common.sacrud_env(fun)
```

```
pyramid_sacrud.common.set_jinja2_silent_none(config)
```

if variable is None print " instead of 'None'

Webhelpers paginator

```
pyramid_sacrud.common.paginator.get_current_page(request)
```

```
pyramid_sacrud.common.paginator.get_paginator(request, items_per_page=10)
```

1.5.4 Views

Views for Pyramid frontend

```
pyramid_sacrud.views.add_global_params(event)
```

```
pyramid_sacrud.views.sa_home(request)
```

Views for CRUD actions

```
class pyramid_sacrud.views.CRUD.Action(request)
```

Bases: `pyramid_sacrud.views.CRUD.CRUD`

```
mass_delete()
```

```
class pyramid_sacrud.views.CRUD.Add(request)
```

Bases: `pyramid_sacrud.views.CRUD.CRUD`

```
sa_add(*args, **kwargs)
```

```
class pyramid_sacrud.views.CRUD.CRUD(request)
```

Bases: `object`

```
flash_message (message, status='success')  
get_response (options, template_attr)  
class pyramid_sacrud.views.CRUD.Delete (request)  
    Bases: pyramid_sacrud.views.CRUD.CRUD  
    sa_delete ()  
class pyramid_sacrud.views.CRUD.List (request)  
    Bases: pyramid_sacrud.views.CRUD.CRUD  
    sa_list (*args, **kwargs)
```

1.5.5 Security

List of permissions

1.5.6 Exceptions

Exceptions for pyramid_sacrud

```
exception pyramid_sacrud.exceptions.SacrudMessagedException (message, status='error')  
    Bases: exceptions.Exception
```

Just raise this exception

```
raise SacrudMessagedException('My Super Message', status='error')
```

status = error|warning|success

Contribute

2.1 Backend contrubute

Same as [contributor's section](#) of the documentation of Pyramid project.

2.2 Frontend contribute

For working with CSS and JavaScript you need install [Node.js](#), [NPM](#), [Bower](#), [Browserify](#) and [Gulp](#).

Note: If you don't have Node.js and NPM installed, get it first.

Installing Node.js dependencies:

```
npm install
```

Install Bower:

```
npm install --global bower
```

Install packages with Bower:

```
bower install
```

Packages will be install to **pyramid_sacrud/bower_components/**

2.2.1 CSS

SS files are located in **pyramid_sacrud/static/css/**.

Before changing css files you may run **gulp watch** task:

```
gulp watch
```

or you can change css and then and run **gulp css** task:

```
gulp css
```

Note: For more information about **gulp css** task see **gulpfile.js**

When you change any **css** file gulp concatenates all files in folder and create `__pyramid_sacrud.css` file in `pyramid_sacrud/static/css/`.

2.2.2 JavaScript

pyramid_sacrud use **Browserify**, file for browserify build is **main.js** in `pyramid_sacrud/static/js/`.

```
pyramid_sacrud/pyramid_sacrud/static/js
|
|   main.js
|   __pyramid_sacrud.js
|   __pyramid_sacrud.js.map
|
+---app
|   |   options.js
|   |
|   /---common
|       |   checkbox.js
|       |   popup.js
|       |   selectable.js
|       |
|   /---tests
|       |   link-check.js
|       |   login-logout.js
|       |   tests.js
```

Before changing **js** modules you may run **gulp watch** task:

```
gulp watch
```

or you can change js and then and run **gulp browserify** task:

```
gulp browserify
```

When you change any js file, browserify build `__pyramid_sacrud.js` in `pyramid_sacrud/static/js/`.

Note: For more information about **gulp browserify** task see **gulpfile.js**

Tip: For more information about browserify visit [Browserify](#)

Project modules

Options

JQuery selectors list.

Popup

Popup object is needed for works with pop-up window.

Create a new Popup:

```
1 var Popup = require('popup.js');
2 var popup = new Popup(options);
```

Arguments:

- options - Options object.

SelectableTable

SelectableTable object is needed for works with table. Using JQuery-UI Selectable widget.

Create a new SelectableTable:

```
1 var SelectableTable = require('selectable.js');
2 var selectable_table = new SelectableTable(el, options);
```

Arguments:

- el - JQuery selector (set in options.popup).
- options - Options object.

Install modules

All modules are installed by using **bower**. After installing, you need set path to module in package.json in “browser” and specify “exports” and “depends”(if need) in “browserify-shim” settings, default **package.json** file looks like:

```
1 "browser": {
2   "jquery": "./bower_components/jquery/dist/jquery.min.js",
3   "jquery-ui": "./bower_components/jquery-ui/ui/minified/jquery-ui.min.js"
4 },
5 "browserify-shim": {
6   "jquery": "$",
7   "jquery-ui": {
8     "depends": "jquery"
9   },
10 }
```

Tip: For more information about browserify visit [Browserify-Shim](#)

For update each dependency in package.json, just use **npm-check-updates**.

```
$ npm install -g npm-check-updates
$ npm-check-updates -u
$ npm install
```

Using modules

To use module, you need to define it in **main.js** via **require()** function:

```
1 require('jquery');
```

After that, they will be available for entire project.

reate a new module

To define a module, just create a JavaScript file **my_module.js**, in **pyramid_sacrud/static/js/** and write something like this:

```
1 module.exports = function some_func(args) {  
2   console.log(args)  
3 };
```

Add it in **main.js** via **require()** function and call, to use in site:

```
1 var myModule = require('my_module');  
2 myModule(args);
```

2.2.3 Testing

Install Testing tools:

```
npm install
```

or install package manually

```
npm install mocha chai --save-dev
```

Tests are written using Mocha framework, you need install [Mocha](#) and [Chai](#).

Note: Also, you need a project running on the **localhost:6543** port. See [example](#) project.

Important: Install Mocha in global (npm install mocha -global) for Windows.

All tests found in directory **pyramid_sacrud/static/js/test/**

To run tests for javascript, use **npm test** command from category containing package.json:

```
npm test
```

or run

```
mocha -b --timeout 5000 pyramid_sacrud/static/js/tests/
```

2.3 Localization contribute

2.3.1 setup.cfg

See basic settings in *setup.cfg* (More information see [there](#))

extract_messages:

```
python setup.py extract_messages
```

update_catalog:

```
python setup.py update_catalog
```

compile_catalog:

```
python setup.py compile_catalog
```

Or just run it through *make*

```
make locale
```

For translate use `_ps` function:

```

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 Internationalization and Localization
11 """
12 from pyramid.i18n import TranslationStringFactory
13
14 _ps = TranslationStringFactory('pyramid_sacrud')
15
16
17 def includeme(config):
18     config.add_translation_dirs('pyramid_sacrud:locale/')
19     config.scan('.views')
```

In templates:

```

<div class="dashboard-title">{{ _ps('Dashboard') }}</div>
{{ _ps(_(crumb.name)) }}
```

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

License

The project is licensed under the MIT license.

Indices and tables

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

p

- `pyramid_sacrud`, [15](#)
- `pyramid_sacrud.breadcrumbs`, [15](#)
- `pyramid_sacrud.common`, [18](#)
- `pyramid_sacrud.common.paginator`, [18](#)
- `pyramid_sacrud.exceptions`, [19](#)
- `pyramid_sacrud.security`, [19](#)
- `pyramid_sacrud.views`, [18](#)
- `pyramid_sacrud.views.CRUD`, [18](#)

A

Action (class in pyramid_sacrud.views.CRUD), 18
Add (class in pyramid_sacrud.views.CRUD), 18
add_global_params() (in module pyramid_sacrud.views), 18

B

breadcrumbs() (in module pyramid_sacrud.breadcrumbs), 15

C

CRUD (class in pyramid_sacrud.views.CRUD), 18

D

Delete (class in pyramid_sacrud.views.CRUD), 19

F

flash_message() (pyramid_sacrud.views.CRUD.CRUD method), 18

G

get_crumb() (in module pyramid_sacrud.breadcrumbs), 18
get_current_page() (in module pyramid_sacrud.common.paginator), 18
get_models_from_settings() (in module pyramid_sacrud.common), 18
get_paginator() (in module pyramid_sacrud.common.paginator), 18
get_response() (pyramid_sacrud.views.CRUD.CRUD method), 19
get_settings_param() (in module pyramid_sacrud.common), 18
get_table() (in module pyramid_sacrud.common), 18
get_table_verbose_name() (in module pyramid_sacrud.common), 18

I

import_from_string() (in module pyramid_sacrud.common), 18

includeme() (in module pyramid_sacrud), 15

L

List (class in pyramid_sacrud.views.CRUD), 19

M

mass_delete() (pyramid_sacrud.views.CRUD.Action method), 18

P

pkg_prefix() (in module pyramid_sacrud.common), 18
preprocessing_value() (in module pyramid_sacrud.common), 18
pyramid_sacrud (module), 15
pyramid_sacrud.breadcrumbs (module), 15
pyramid_sacrud.common (module), 18
pyramid_sacrud.common.paginator (module), 18
pyramid_sacrud.exceptions (module), 19
pyramid_sacrud.security (module), 19
pyramid_sacrud.views (module), 18
pyramid_sacrud.views.CRUD (module), 18

S

sa_add() (pyramid_sacrud.views.CRUD.Add method), 18
sa_delete() (pyramid_sacrud.views.CRUD.Delete method), 19
sa_home() (in module pyramid_sacrud.views), 18
sa_list() (pyramid_sacrud.views.CRUD.List method), 19
sacrud_env() (in module pyramid_sacrud.common), 18
SacrudMessageException, 19
set_jinja2_silent_none() (in module pyramid_sacrud.common), 18