Mezzanine Page Auth Documentation

Release 0.1

Simone Dalla

November 18, 2013

Contents

Mezzanine Page Auth is Mezzanine module for add group-level permission to Pages.

CHAPTER 1

Installation

The easiest method is to install directly from pypi using pip by running the command below, which will also install the required dependencies mentioned above:

```
$ pip install mezzanine-auth-pages
```

If you prefer, you can download mezzanine-auth-pages and install it directly from source:

```
$ python setup.py install
```

Add `mezzanine_page_auth` to your `INSTALLED_APPS` setting before all mezzanine apps:

```
INSTALLED_APPS = (
    # ...
    'mezzanine_page_auth',
    'mezzanine.boot',
    'mezzanine.conf',
    'mezzanine.core',
    # ...
)
```

You will then want to create the necessary tables. If you are using South for schema migrations, you'll want to:

```
$ python manage.py migrate mezzanine_page_auth
```

1.1 Middleware

Enable `PageAuthMiddleware` middleware in your settings module as follows:

```
MIDDLEWARE_CLASSES = (
    # ...
    "mezzanine.core.middleware..."
    "mezzanine.pages.middleware..."
    "mezzanine_page_auth.middleware.PageAuthMiddleware",
    # ...
)
```

The order of `MIDDLEWARE_CLASSES` is important. You should include the `PageAuthMiddleware` middleware after other Mezzanine middlewares in the list.

1.2 Template Context Processors

Enable `page_auth` template context processors in your settings module as follows:

```
TEMPLATE_CONTEXT_PROCESSORS = (
    # ...
    'mezzanine_page_auth.context_processors.page_auth',
)
```

The order of `MIDDLEWARE_CLASSES` is important. You should include the `PageAuthMiddleware` middleware after other Mezzanine middlewares in the list.

1.3 Mezzanine Settings

Configure `EXTRA_MODEL_FIELDS` Mezzanine setting in your settings module as follows:

```
EXTRA_MODEL_FIELDS = (
    # ...
    (
        "mezzanine.pages.models.Page.groups",
        "ManyToManyField",
        ("auth.Group",),
        {"blank": True, "null": True, "verbose_name": 'groups',
         'symmetrical': False, 'through': "mezzanine_page_auth.PageAuthGroup"},
    ),
    # ...
)
```

for inject field `groups` into Mezzanine's models (reference to Field Injection Mezzanine documantation).

CHAPTER 2

Templates

If you don't want display the pages into Page Menus you can override the page menu templates (`tree.html`, `dropdown.html`, `footer.html`) checking if the `pk` of current page is not in `unauthorized_pages` template context variable.

Here's a example of customization of original `tree.html` template:

```
# ...
{% if page.in_menu and page.pk not in unauthorized_pages %}
  <li class="
            {% if page.is_current %} active{% endif %}
            {% if not page.is_primary and forloop.first %} first{% endif %}
            {% if forloop.last %} last{% endif %}"
      id="tree-menu-{{ page.html_id }}">
    <a href="{{ page.get_absolute_url }}">{{ page.title }}</a>
    {# wrap the next line with 'if page.is_current_or_ascendant' #}
    {# to only show child pages in the menu for the current page #}
    {% if page.is_current_or_ascendant %}
        {% if page.has_children_in_menu %}{% page_menu page %}{% endif %}
    {% endif %}
  </li>
{% endif %}
# ...
```

The `unauthorized_pages` variable is inserted into template context by context processor `'mezzanine_page_auth.context_processors.page_auth'` (reference *Template Context Processors*)

CHAPTER 3

Links

- Documentation: http://mezzanine_page_auth.readthedocs.org/
- Official repository: https://github.com/simodalla/mezzanine_page_auth/
- Package: https://pypi.python.org/pypi/mezzanine-page-auth/

CHAPTER 4

License

Mezzanine Page Auth is BSD licensed. See the `LICENSE` file in the top distribution directory for the full license text.