# Contents

1 User’s Guide 3
   1.1 Installation 3
   1.2 Usage 3

2 API Reference 5
   2.1 API Docs 5

3 Additional Notes 9
   3.1 Contributing 9
   3.2 Changes 11
   3.3 License 11
   3.4 Authors 12

Python Module Index 13
Static pages module for Invenio.

*This is an experimental developer preview release.*

- Free software: GPLv2 license
- Documentation: https://invenio-pages.readthedocs.io/
This part of the documentation will show you how to get started in using Invenio-Pages.

**Installation**

**Usage**

Static pages module for Invenio.
API Reference

If you are looking for information on a specific function, class or method, this part of the documentation is for you.

**API Docs**

Static pages module for Invenio.

```python
class invenio_pages.ext.InvenioPages(app=None)
    Invenio-Pages extension.
    Extension initialization.
    
    Parameters **app** -- The Flask application. (Default: None)
    
    init_app(app)
    Flask application initialization.
    
    Parameters **app** -- The Flask application.
    
    Returns The `invenio_pages.ext.InvenioPages` instance initialized.
    
    init_config(app)
    Initialize configuration.
    
    Parameters **app** -- The Flask application.
    
    static wrap_errorhandler(app)
    Wrap error handler.
    
    Parameters **app** -- The Flask application.
```

**Configuration**

Page module config.
invenio-pages Documentation, Release 1.0.0a4

invenio_pages.config.PAGES_DEFAULT_TEMPLATE = 'invenio_pages/default.html'
Default template to render.

invenio_pages.config.PAGES_TEMPLATES = [('invenio_pages/default.html', 'Default'), ('invenio_pages/dynamic.html', 'Default dynamic')]
List of available templates for pages.

invenio_pages.config.PAGES_WHITELIST_CONFIG_KEYS = ['THEME_SITENAME']
List of configuration variables accessible in the dynamic pages.

Models

Pages module models.

class invenio_pages.models.Page(**kwargs)
    Represents a page.
    A simple constructor that allows initialization from kwargs.
    Sets attributes on the constructed instance using the names and values in kwargs.
    Only keys that are present as attributes of the instance’s class are allowed. These could be, for example, any mapped columns or relationships.

    content
        Page content. Default is pages/templates/default.html

    description
        Page description.

classmethod get_by_url(url)
    Get a page by URL.
    Parameters
        url – The page URL.
    Returns

    id
        Page identifier.

    template_name
        Page template name.

    title
        Page title.

    url
        Page url.

    validate_template_name(key, value)
        Validate template name.
        Parameters
            • key – The template path.
            • value – The template name.
        Raises
            ValueError – If template name is wrong.

class invenio_pages.models.PageList(**kwargs)
    Represent association between page and list.
    A simple constructor that allows initialization from kwargs.
    Sets attributes on the constructed instance using the names and values in kwargs.
Only keys that are present as attributes of the instance’s class are allowed. These could be, for example, any mapped columns or relationships.

**id**
PageList identifier.

**list**
Relation to the list.

**list_id**
Id of a list.

**page**
Relation to the page.

**page_id**
Id of a page.

## Views

Views for Pages module.

**invenio_pages.views.handle_not_found(exception, **extra)**
Custom blueprint exception handler.

**invenio_pages.views.preload_pages()**
Register all pages before the first application request.

**invenio_pages.views.render_page(path)**
Internal interface to the page view.

- **Parameters** `path` – Page path.
- **Returns** The rendered template.

**invenio_pages.views.render_string(source)**
Render a string in sandboxed environment.

- **Parameters** `source` – A string containing the page source.
- **Returns** The rendered template.

**invenio_pages.views.view()**
Public interface to the page view.

Models: `pages.pages`. Templates: Uses the template defined by the template_name field or `pages/default.html` if template_name is not defined. Context: page `pages.pages` object.
Notes on how to contribute, legal information and changes are here for the interested.

**Contributing**

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

**Types of Contributions**

**Report Bugs**


If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

**Fix Bugs**

Look through the GitHub issues for bugs. Anything tagged with “bug” is open to whoever wants to implement it.

**Implement Features**

Look through the GitHub issues for features. Anything tagged with “feature” is open to whoever wants to implement it.
Write Documentation

Invenio-Pages could always use more documentation, whether as part of the official Invenio-Pages docs, in docstrings, or even on the web in blog posts, articles, and such.

Submit Feedback

The best way to send feedback is to file an issue at https://github.com/inveniosoftware/invenio-pages/issues.

If you are proposing a feature:

• Explain in detail how it would work.
• Keep the scope as narrow as possible, to make it easier to implement.
• Remember that this is a volunteer-driven project, and that contributions are welcome :)

Get Started!

Ready to contribute? Here’s how to set up invenio for local development.

1. Fork the invenio repo on GitHub.
2. Clone your fork locally:

   ```
   $ git clone git@github.com:your_name_here/invenio-pages.git
   ```
3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

   ```
   $ mkvirtualenv invenio-pages
   $ cd invenio-pages/
   $ pip install -e .[all]
   ```
4. Create a branch for local development:

   ```
   $ git checkout -b name-of-your-bugfix-or-feature
   ```

   Now you can make your changes locally.
5. When you’re done making changes, check that your changes pass tests:

   ```
   $ ./run-tests.sh
   ```

   The tests will provide you with test coverage and also check PEP8 (code style), PEP257 (documentation), flake8 as well as build the Sphinx documentation and run doctests.
6. Commit your changes and push your branch to GitHub:

   ```
   $ git add .
   $ git commit -s -m "Your detailed description of your changes."
   $ git push origin name-of-your-bugfix-or-feature
   ```
7. Submit a pull request through the GitHub website.
Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests and must not decrease test coverage.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring.
3. The pull request should work for Python 2.7, 3.3, 3.4 and 3.5. Check https://travis-ci.com/inveniosoftware/invenio-pages/pull_requests and make sure that the tests pass for all supported Python versions.

Changes

Version 1.0.0a4 (released 2017-08-18):

- Update minimum Flask version to 0.11.1.
- Improve documentation and examples.

Version 1.0.0a3 (released 2016-06-15):

- Major refactoring for Invenio 3.
- Adds versioning support for pages.

Version 0.1.2 (released 2015-10-07):

- Removes calls to PluginManager consider_setuptools_entrypoints() removed in PyTest 2.8.0.
- Adds missing `invenio_base` dependency.

Version 0.1.1 (released 2015-08-25):

- Adds missing `invenio_upgrader` dependency and amends past upgrade recipes following its separation into standalone package.
- Overrides default wtforms field for content column in order to display it properly. (closes inveniosoftware/invenio#3311)

Version 0.1.0 (released 2015-07-22):

- Initial public release.

License

Invenio is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.
Invenio is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with Invenio; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307, USA.

In applying this license, CERN does not waive the privileges and immunities granted to it by virtue of its status as an Intergovernmental Organization or submit itself to any jurisdiction.

Authors

Static pages module for Invenio.

- Adrian Baran <bouzlibop@gmail.com>
- Alizée Pace <alizee.pace@gmail.com>
- Eirini Psallida <eirini.psallida@cern.ch>
- Javier Martín Montull <javier.martin.montull@cern.ch>
- Jiri Kuncar <jiri.kuncar@cern.ch>
- Lars Holm Nielsen <lars.holm.nielsen@cern.ch>
- Leonardo Rossi <leonardo.r@cern.ch>
- Sami Hiltunen <sami.mikael.hiltunen@cern.ch>
- Tibor Simko <tibor.simko@cern.ch>
- Yoan Blanc <yoan.blanc@cern.ch>
Python Module Index

i
invenio_pages, 3
invenio_pages.config, 5
invenio_pages.ext, 5
invenio_pages.models, 6
invenio_pages.views, 7
C
content (invenio_pages.models.Page attribute), 6

D
description (invenio_pages.models.Page attribute), 6

G
get_by_url() (invenio_pages.models.Page class method), 6

H
handle_not_found() (in module invenio_pages.views), 7

I
id (invenio_pages.models.Page attribute), 6
id (invenio_pages.models.PageList attribute), 7
init_app() (invenio_pages.ext.InvenioPages method), 5
init_config() (invenio_pages.ext.InvenioPages method), 5
invenio_pages (module), 3
invenio_pages.config (module), 5
invenio_pages.ext (module), 5
invenio_pages.models (module), 6
invenio_pages.views (module), 7
InvenioPages (class in invenio_pages.ext), 5

L
list (invenio_pages.models.PageList attribute), 7
list_id (invenio_pages.models.PageList attribute), 7

P
Page (class in invenio_pages.models), 6
page (invenio_pages.models.PageList attribute), 7
page_id (invenio_pages.models.PageList attribute), 7
PageList (class in invenio_pages.models), 6
PAGES_DEFAULT_TEMPLATE (in module invenio_pages.config), 5
PAGES_WHITELIST_CONFIG_KEYS (in module invenio_pages.config), 6
preload_pages() (in module invenio_pages.views), 7

R
render_page() (in module invenio_pages.views), 7
render_string() (in module invenio_pages.views), 7

T
template_name (invenio_pages.models.Page attribute), 6
title (invenio_pages.models.Page attribute), 6

U
url (invenio_pages.models.Page attribute), 6

V
validate_template_name() (invenio_pages.models.Page method), 6
view() (in module invenio_pages.views), 7

W
wrap_errorhandler() (invenio_pages.ext.InvenioPages static method), 5