STANDING ON THE SHOULDERS OF GIANTS

THE KOTTI WEB APPLICATION FRAMEWORK

ANDREAS KAISER

Owner & CTO of Xo7 GmbH | Willich, Germany (next to Düsseldorf)

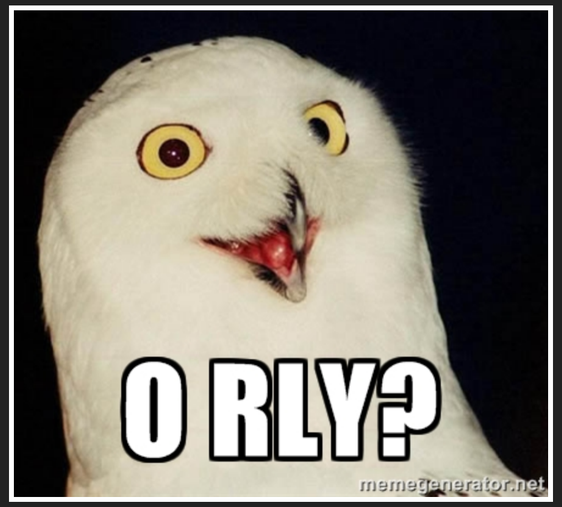
@diskokaiser | disko | irc://irc.freenode.net/#kotti

THIS TALK

- Why does Kotti exist?
- Who are these giants?
- How are they utilized by Kotti?
- Example / Code
- Q&A (if we have time)

WHY?

YET ANOTHER WEB FRAMEWORK?



FEATURES

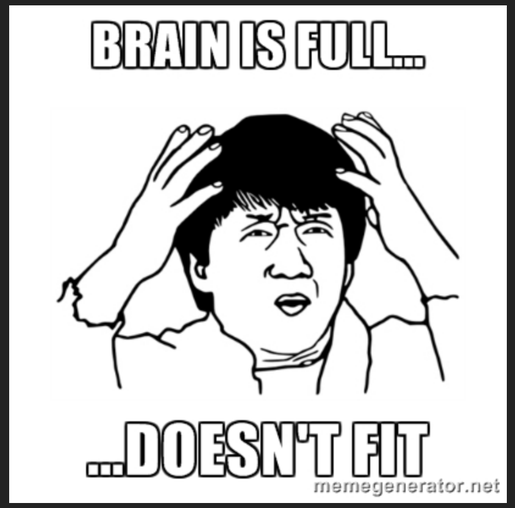
- full featured CMS
- lots of add ons (varying quality)
- OFS (object file system)
- security (permissions, roles, groups)
- workflows

Can be a perfect choice when it fits your needs

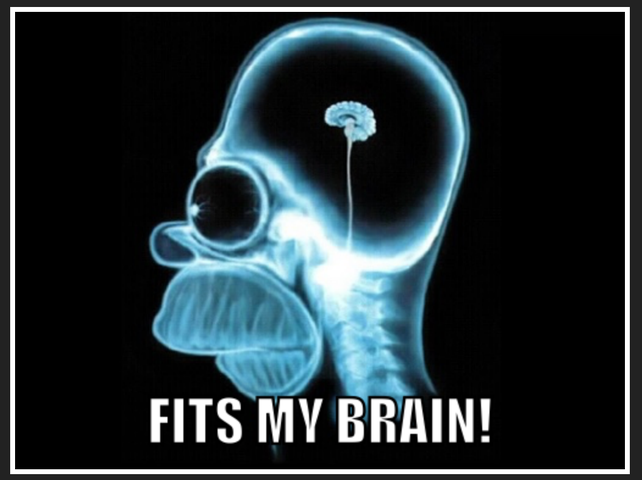
BUT

does **not** fit **all** kinds of applications.

- multiple competing technologies
- doesn't conform to "The Zen of Python"
- complex


BRAIN IS FULL...
...DOESN'T FIT
memegenerator.net


Pyramid™


FITS MY BRAIN!

FEATURES

- small core
- excellent documentation
- pythonic
- low level (micro framework)
- unopinionated
  - persistence
  - templating / forms
  - authentication / authorization sources
- *“framework framework”*

CONCLUSION

- only provides stuff we need
- doesn't come with unneeded ballast
- no need to
  “waste time fighting the framework's decisions”
- perfect foundation!

MAKE SOME CHOICES!

- persistence
- traversal or URL dispatch
- templating & forms
- authentication & authorization sources

- probably the most advanced ORM for Python
- database agnostic
- has many nice, useful features
  - hybrid properties
  - association proxies
  - ordering list
- transaction integration with pyramid

THE NODE CLASS

- adjacency list pattern
  - parent
  - children
- single root node => node tree
- dictionary protocol

THE NODE CLASS

DICTIONARY PROTOCOL

```
from kotti.resources import Document
from kotti.resources import get_root

root = get_root()

root['about']
<Document 2 at /about>

root['my-document'] = Document(title='My Document', description='foo',
                               body='<p>some HTML</p>')
```

THE NODE CLASS

TRAVERSAL

```
a = root['a'] = Document(title='A', description='Document A')
b = root['a']['b'] = Document(title='B', description='Document B')
c = root['a']['b']['c'] = Document(title='C', description='Document C')
```

Object	URL
a	/a
b	/a/b
c	/a/b/c

POLIMORPHIC QUERIES

```
from kotti.resources import get_root
from kotti.resources import Node

root = get_root()
print root.children:
    print(type(c))

"<class 'kotti.resources.Document'>"
"<class 'kotti.resources.File'>"

print Node.query.filter(Node.title == 'My Document').one()
"<Document 5 at /my-document>"
```

JOINED TABLE INHERITANCE

- class hierarchy is broken up among dependent tables
- each class represented by its own table
- the respective table only includes attributes local to that class

EVENTS

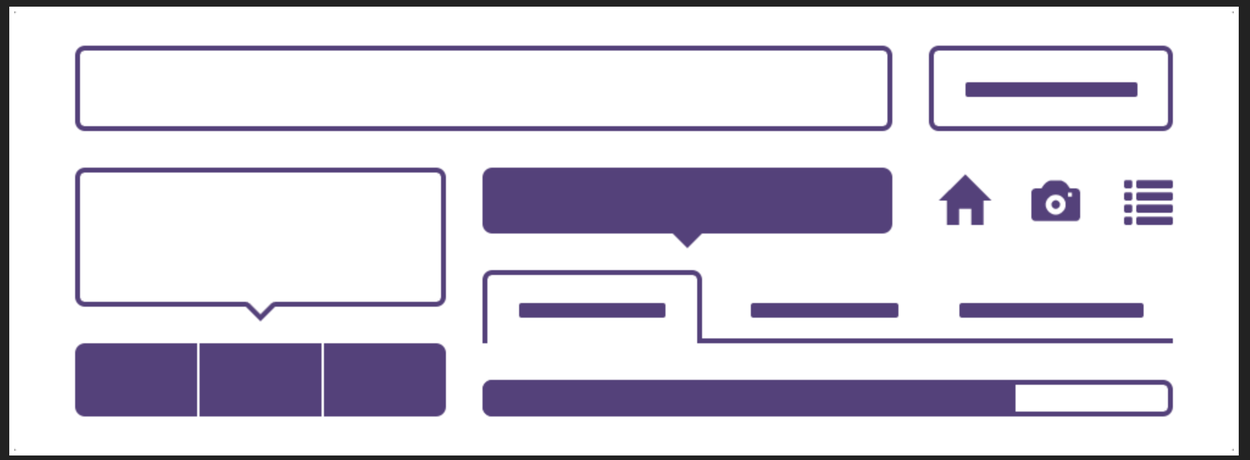
- before_flush
  - ObjectUpdate
  - ObjectInsert
  - ObjectDelete

ALEMBIC

- DB migrations
- DDL (transactional, if supported by DBMS)
- DML
- environments

Bootstrap

- has all the components for modern UIs
- responsive
- well known
- easy to customize



Colander & Deform

COLANDER

- define data schema
- validate & deserialize
  - HTML forms
  - JSON
  - XML
- serialize Python structures to
  - strings
  - mappings
  - lists

DEFORM

- render HTML forms from structures serialized by Colander
- outputs Bootstrap 3 forms (Deform 2)

- a content workflow system
- **states** define
  - role / permission mapping
- **transitions** define
  - from_state
  - to_state
  - required permission

- storing and serving files in web applications
- multiple backends
  - local filesystem
  - S3
  - GridFS
  - roll your own
- integrates with SQLAlchemy
  - files are handled like a plain model attribute
  - transaction aware

WIRING IT ALL TOGETHER…

- started by Daniel Nouri in 2011
- BSD licensed
- 1.0.0 in January 2015
- current version: 1.1.4
- 9k downloads per month
- still small, but active & healthy community
- contributions are always welcome

CODE QUALITY

a.k.a. "everyone loves badges"

almost Heisenberg quality

continuous integration (Python 2.6, 2.7, PostgreSQL, MySQL, SQLite)

static code analysis (Codacy, Code Climate, QuantifiedCode)

(except 1 testing requirement)

CONFIGURATION

```
[app:kotti]
use = egg:kotti
sqlalchemy.url = sqlite:///%(here)s/Kotti.db
# sqlalchemy.url = postgres://user:pass@host/db
kotti.configurators =
    kotti_tinymce.kotti_configure
    kotti_youraddon.kotti_configure

[filter:fanstatic]
use = egg:fanstatic#fanstatic

[pipeline:main]
pipeline =
    fanstatic
    kotti

[server:main]
use = egg:waitress#main
host = 127.0.0.1
port = 5000
```

EXAMPLE OPTIONS

Option	Purpose
kotti.available_types	List of active content types
kotti.configurators	List of advanced functions for config
kotti.root_factory	Override Kotti’s default Pyramid root factory
kotti.populators	List of functions to fill initial database
kotti.search_content	Override Kotti’s default search function
kotti.asset_overrides	Override Kotti’s templates
kotti.authn_policy_factory	Component used for authentication
kotti.authz_policy_factory	Component used for authorization

EXAMPLE OPTIONS (CONTINUED)

Option	Purpose
kotti.caching_policy_chooser	Component for choosing the cache header policy
kotti.url_normalizer	Component used for url normalization
kotti.max_file_size	Max size for file uploads
kotti.depot.*.*	Configure the blob storage
kotti.sanitizers	Configure available sanitizers
kotti.sanitize_on_write	Configure sanitizers to be used on write access to resource objects

SECURITY

- use SQLAlchemy to…
  - store pricipals (users & groups) in the DB
  - attach (inheritable) ACLs to each node
- use Pyramid for…
  - authentication
  - authorization
- use repoze.workflow to…
  - recompute ACLs on workflow state changes

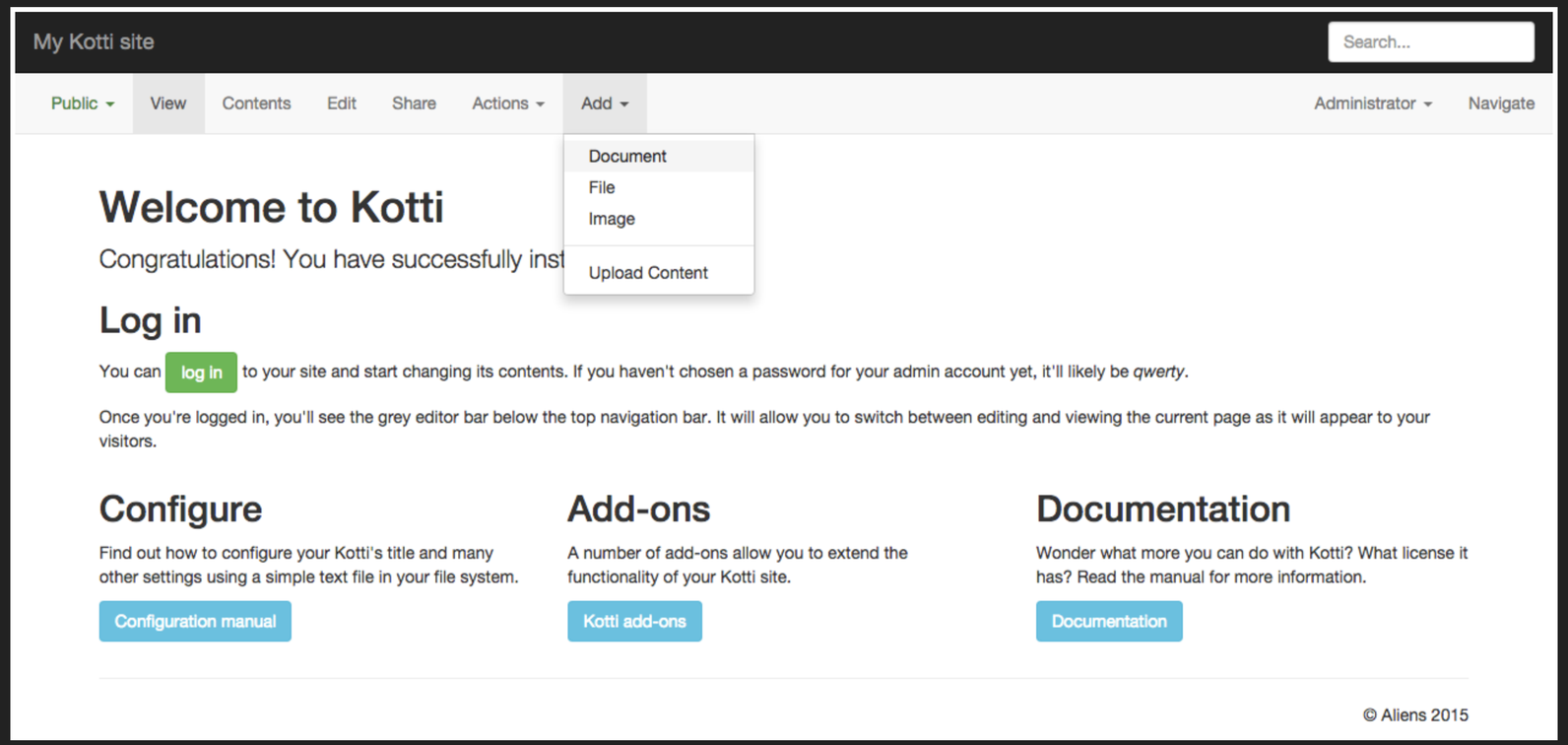
EXAMPLE

CREATING AN ADDON

```
$ pcreate -s kotti kotti_myaddon
Author name [Andreas Kaiser]:
Author email [disko@binary-punks.com]:
Github username [Kotti]:
[… lot of output …]
================================================================================
Welcome to Kotti!

Documentation: http://kotti.readthedocs.org/
Development:   https://github.com/Kotti/Kotti/
Issues:        https://github.com/Kotti/Kotti/issues?state=open
IRC:           irc://irc.freenode.net/#kotti
Mailing List:  https://groups.google.com/group/kotti
================================================================================
```

CUSTOM CONTENT TYPE

```
from kotti.resources import Content
from sqlalchemy import *

class Document(Content):

    id = Column(Integer(),
                ForeignKey('contents.id'),
                primary_key=True)

    body = Column(UnicodeText())
    mime_type = Column(String(30))

    type_info = Content.type_info.copy(
        name=u'Document',
        title=_(u'Document'),
        add_view=u'add_document',
        addable_to=[u'Document'])
```

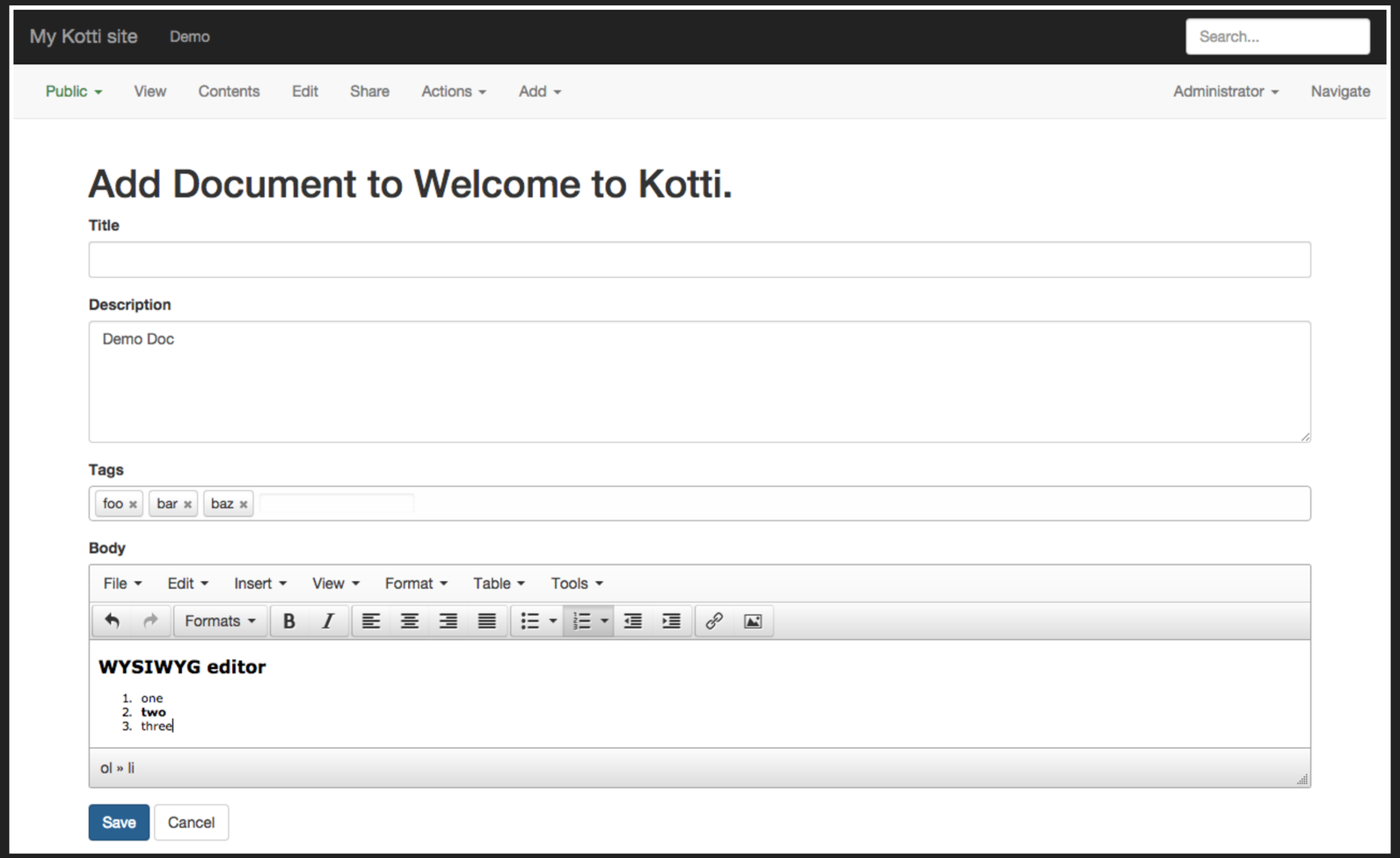

My Kotti site
Search...
Public
View
Contents
Edit
Share
Actions
Add
Administrator
Navigate
Document
File
Image
Upload Content
Welcome to Kotti
Congratulations! You have successfully inst
Log in
You can log in to your site and start changing its contents. If you haven't chosen a password for your admin account yet, it'll likely be qwerty.
Once you're logged in, you'll see the grey editor bar below the top navigation bar. It will allow you to switch between editing and viewing the current page as it will appear to your visitors.
Configure
Find out how to configure your Kotti's title and many other settings using a simple text file in your file system.
Configuration manual
Add-ons
A number of add-ons allow you to extend the functionality of your Kotti site.
Kotti add-ons
Documentation
Wonder what more you can do with Kotti? What license it has? Read the manual for more information.
Documentation
© Aliens 2015

SCHEMA DEFINITION FOR VALIDATION AND FORM CREATION

```
import colander
import deform
from kotti.views.edit.content import ContentSchema

class DocumentSchema(ContentSchema):
    body = colander.SchemaNode(
        colander.String(),
        title=_(u'Body'),
        widget=deform.widget.RichTextWidget(),
        missing=u"")
```

ADD / EDIT FORMS

```
from kotti.resources import Document
from kotti.views.form import AddFormView
from kotti.views.form import EditFormView
from pyramid.view import view_config

@view_config(name=Document.type_info.add_view, permission='add',
             renderer='kotti:templates/edit/node.pt')
class DocumentAddForm(AddFormView):
    schema_factory = DocumentSchema
    add = Document
    item_type = _(u"Document")

@view_config(context=Document, name='edit', permission='edit',
             renderer='kotti:templates/edit/node.pt')
class DocumentEditForm(EditFormView):
    schema_factory = DocumentSchema
```


My Kotti site
Demo
Search...
Public
View
Contents
Edit
Share
Actions
Add
Administrator
Navigate
Add Document to Welcome to Kotti.
Title
Description
Demo Doc
Tags
foo
bar
baz
Body
File
Edit
Insert
View
Format
Table
Tools
Formats
WYSIWYG editor
one
two
three
ol » li
Save
Cancel

Add Document to Welcome to Kotti.

There was a problem with your submission

Title

Required

Description

Demo Doc

Tags

foo × bar × baz ×

Body

File Edit Insert View Format Table Tools

Formats

WYSIWYG editor

1. one
2. **two**
3. three

h2

Save Cancel

VIEW(S)

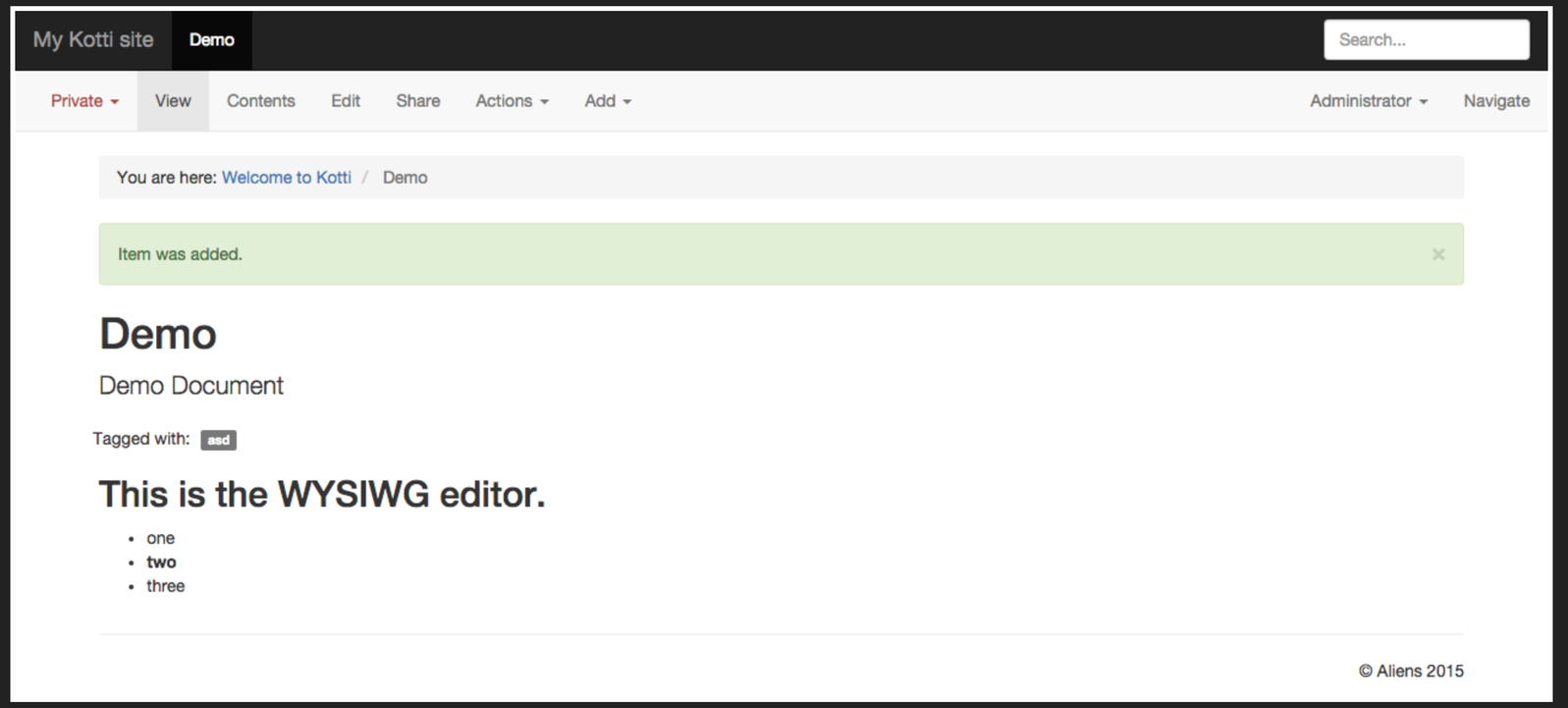
```
from pyramid.view import view_config
@view_config(name='view', context=Document, permission='view',
             renderer='kotti:templates/view/document.pt')
def document_view(context, request):
    return {}
```

OR

```
from pyramid.view import view_config
from pyramid.view import view_defaults
from kotti.views import BaseView

@view_defaults(context=Document, permission='view')
class DocumentViews(BaseView):
    @view_config(name='view', renderer='kotti:templates/view/document.pt')
    def view(self):
        return {}

    @view_config(name='view2', renderer='kotti:templates/view/document2.pt')
    def view(self):
        return {'answer': 42}

    @view_config(name='json', renderer='json')
    def view(self):
        return {'title': self.context.title, 'body': self.context.body, ...}
        # return self.context
```

TEMPLATE(S)

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml"
      metal:use-macro="api.macro('kotti:templates/view/master.pt')">
  <article metal:fill-slot="content" class="document-view content">
    <h1>${context.title}</h1>
    <p class="lead">
      ${context.description}
    </p>
    <div tal:replace="api.render_template('kotti:templates/view/tags.pt')" />
    <div class="body" tal:content="structure context.body | None">
    </div>
  </article>
</html>
```


My Kotti site
Demo
Search...
Private
View
Contents
Edit
Share
Actions
Add
Administrator
Navigate
You are here: Welcome to Kotti / Demo
Item was added.
Demo
Demo Document
Tagged with: asd
This is the WYSIWG editor.
one
two
three
© Aliens 2015

THE FUTURE

- will always stay “lean and mean in all of the right ways”
- Python 3 support

THANK YOU!

QUESTIONS?