Application architecture

Lars Holm Nielsen
CERN

Flask 101

- Blueprint
- Extension
- Application

- Application context
- Request context

```python
from flask import Blueprint, Flask, request

# Blueprint
bp = Blueprint('bp', __name__)

@bp.route('/')
def my_user_agent():
    # Executing inside request context
    return request.headers['User-Agent']

# Extension
class MyExtension(object):
    def __init__(self, app=None):
        if app:
            self.init_app(app)

    def init_app(self, app):
        app.config.setdefault('MYCONF', True)

# Application
app = Flask(__name__)
ext = MyExtension(app)
app.register_blueprint(bp)
```

Application assembly

- Application factory
- Assembly phases
  - 1. Application creation
  - **2. Configuration loading**
  - 3. URL converter loading
  - **4. Flask extension loading**
  - **5. Flask blueprint loading**

```python
from flask import Flask, Blueprint

# Module 1
bp1 = Blueprint(__name__, 'bp1')
@bp1.route('/')
def hello():
    return 'Hello'

# Module 2
bp2 = Blueprint(__name__, 'bp2')
@bp2.route('/')
def world():
    return 'World'

# Application factory
def create_app():
    app = Flask(__name__)
    app.register_blueprint(bp1)
    app.register_blueprint(bp2)
    return app
```

Application assembly

setup.py

```
entry_points={
    'console_scripts': [
        'my-site = invenio_app.cli:cli',
    ],
    'invenio_base.apps': [
        'my_site_records = my_site.records:Mys
    ],
    'invenio_base.api_apps': [
        'my_site = my_site.records:Mysite',
    ],
    'invenio_base.blueprints': [
        'my_site = my_site.theme.views:bluepri
        'my_site_records = my_site.records.vie
    ],
    'invenio_config.module': [
        'my_site = my_site.config',
    ],
```

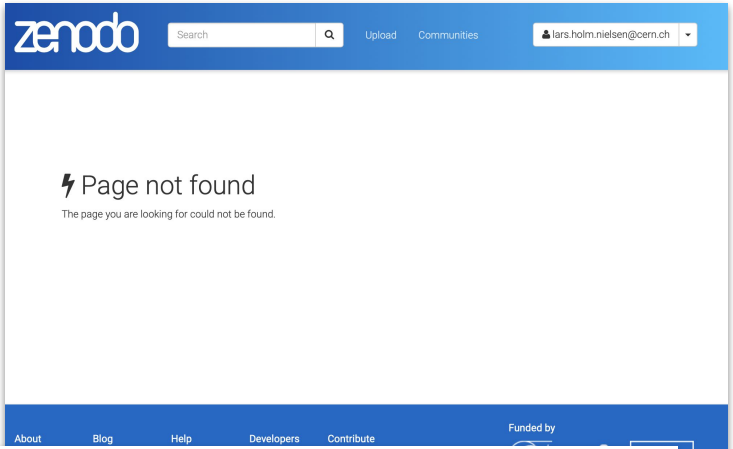
my_site/theme/views.py

```
from flask import Blueprint

blueprint = Blueprint(
    'my_site',
    __name__,
    template_folder='templates',
    static_folder='static',
)
```

Applications and app interfaces

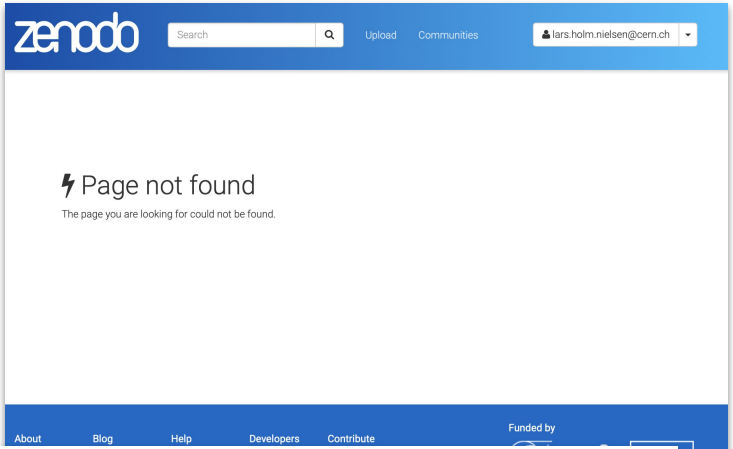
- Applications: UI and API
- Interfaces:
  - WSGI (application server)
  - CLI
  - Celery




```
{
  message: "The requested URL was not found on the server.  If
  you entered the URL manually please check your spelling and try
  again.",
  status: 404
}
```

Applications and app interfaces

- Applications: UI and API
- Interfaces:
  - WSGI (application server)
  - CLI
  - Celery




```
{
  message: "The requested URL was not found on the server.  If
  you entered the URL manually please check your spelling and try
  again.",
  status: 404
}
```

Internals: Invenio-App

invenio_app/factory.py

```
create_api = create_app_factory(
    'invenio',
    config_loader=config_loader,
    blueprint_entry_points=['invenio_base.api_blueprints'],
    extension_entry_points=['invenio_base.api_apps'],
    converter_entry_points=['invenio_base.api_converters'],
    wsgi_factory=wsgi_proxyfix(),
    instance_path=instance_path,
    app_class=app_class(),
)
"""Flask application factory for Invenio REST API."""

create_ui = create_app_factory(
    'invenio',
    config_loader=config_loader,
    blueprint_entry_points=['invenio_base.blueprints'],
    extension_entry_points=['invenio_base.apps'],
    converter_entry_points=['invenio_base.converters'],
    wsgi_factory=wsgi_proxyfix(),
    instance_path=instance_path,
    static_folder=static_folder,
    app_class=app_class(),
)
```

invenio_app/wsgi_ui.py

```
from .factory import create_ui

#: WSGI application for Invenio UI.
application = create_ui()
```

my-site/docker/uwsgi/uwsgi_ui.ini

```
[uwsgi]
socket = 0.0.0.0:5000
module = invenio_app.wsgi_ui:application
master = true
die-on-term = true
processes = 2
threads = 2
```

Questions?