
baka-framework Documentation

Release 0.4.0

Nanang Suryadi

Apr 07, 2018

Contents

1	Petunjuk Penggunaan	3
1.1	Installation	3
1.2	Quickstart	5
2	Referensi API	11
3	Catatan Tambahan	13
3.1	Baka Changelog	13
3.2	License	14
3.3	Cara berkontribusi di Baka framework	15



Selamat datang di Baka framework documentation. Dokumentasi ini terbagi atas bagian yang berbeda. Saya sarankan Anda memulai install dan kemudian menuju ke [Quickstart](#). Selain quickstart, ada juga yang lebih detail tutorial itu menunjukkan bagaimana membuat aplikasi lengkap (walaupun kecil) dengan Baka. jika Anda lebih suka menyelam lebih dalam, periksa dokumentasi api. Pola umum dijelaskan di patterns.

Baka bergantung pada beberapa libraries: [Pyramid](#) untuk core web framework engine dan [Werkzeug](#) WSGI toolkit untuk development server. Librari tersebut tidak terdokumentasi disini. Jika ingin mempelajari lebih lanjut, silakan cek halaman link ini:

- [Pyramid Documentation](#)
- [Werkzeug Documentation](#)

Petunjuk Penggunaan

Document bagian ini yang penting dari semua bagian untuk memulai, usahakan mengikuti langkah-langkah satu per satu instruksi dari development web dengan Baka framework.

1.1 Installation

1.1.1 Python Version

Kita merekomendasikan menggunakan versi Python 3 yang terbaru. Baka supports Python 3.4 dan terbaru, Python 2.7, dan PyPy.

1.1.2 Dependencies

These distributions will be installed automatically when installing Baka.

- [Werkzeug](#) implements WSGI, the standard Python interface between applications and servers.
- [Pyramid](#) is a web framework python.

Optional dependencies

These distributions will not be installed automatically. Baka will detect and use them if you install them.

1.1.3 Virtual environments

Use a virtual environment to manage the dependencies for your project, both in development and in production.

What problem does a virtual environment solve? The more Python projects you have, the more likely it is that you need to work with different versions of Python libraries, or even Python itself. Newer versions of libraries for one project can break compatibility in another project.

Virtual environments are independent groups of Python libraries, one for each project. Packages installed for one project will not affect other projects or the operating system's packages.

Python 3 comes bundled with the `venv` module to create virtual environments. If you're using a modern version of Python, you can continue on to the next section.

If you're using Python 2, see *Install `virtualenv`* first.

Create an environment

Create a project folder and a `venv` folder within:

```
mkdir myproject
cd myproject
python3 -m venv venv
```

On Windows:

```
py -3 -m venv venv
```

If you needed to install `virtualenv` because you are on an older version of Python, use the following command instead:

```
virtualenv venv
```

On Windows:

```
\Python27\Scripts\virtualenv.exe venv
```

Activate the environment

Before you work on your project, activate the corresponding environment:

```
. venv/bin/activate
```

On Windows:

```
venv\Scripts\activate
```

Your shell prompt will change to show the name of the activated environment.

1.1.4 Install Baka

Within the activated environment, use the following command to install Baka:

```
pip install Baka
```

1.1.5 Install virtualenv

If you are using Python 2, the venv module is not available. Instead, install [virtualenv](#).

On Linux, virtualenv is provided by your package manager:

```
# Debian, Ubuntu
sudo apt-get install python-virtualenv

# CentOS, Fedora
sudo yum install python-virtualenv

# Arch
sudo pacman -S python-virtualenv
```

If you are on Mac OS X or Windows, download [get-pip.py](#), then:

```
sudo python2 Downloads/get-pip.py
sudo python2 -m pip install virtualenv
```

On Windows, as an administrator:

```
\Python27\python.exe Downloads\get-pip.py
\Python27\python.exe -m pip install virtualenv
```

Now you can continue to [Create an environment](#).

1.2 Quickstart

Kesulitan dalam mencoba? Halaman ini memperlihatkan betapa mudahnya menjalankan Baka framework. Contoh ini dijalankan dengan berasumsi anda sudah melakukan instalasi Baka framework. Jika belum bisa kehalaman ini [Installation](#) section.

1.2.1 Konfigurasi Sederhana

Baka Aplikasi sangat sederhana seperti flask-like

```
from Baka import Baka
app = Baka(__name__)

@app.route('/')
```

```
def hello_world():
    return 'Hello, World!'
```

Kamu juga dapat menggunakan baka framework dengan sangat sederhana seperti route handler function, misalnya.

```
from baka import Baka
from baka.log import log

app = Baka(__name__)

# route method
@app.route('/')
def index_page(req):
    log.info(req)
    return {'Baka': 'Hello World!'}

@app.route('/home')
def home_page(req):
    log.info(req)
    return {'Route': 'home'}

# root resources routes
class ResourcesPage(object):
    def __init__(self, request):
        self._name = 'Resource Page'
        log.info(request.params)

# GET resource method
@ResourcesPage.GET()
def resources_page_get(root, request):
    return {
        'hello': 'Get Hello resources from Page root %s ' % page._
        ↪name
    }
```

1.2.2 Modular Package/Folder

Dengan menggunakan `baka.include(callable)`, kamu dapat menggabungkan module terpisah dari beberapa file didalam *package module*.

contoh file: `testbaka/view_user.py`

```
from .app import app
```

```
@app.route('/users')
def user(req):
    return {'users': 'all data'}

def includeme(config):
    pass
```

file: testbaka/app.py

```
from baka import Baka
from baka.log import log

app = Baka(__name__)
app.include('testbaka.view_user') # include module dari file view_
    ↳user.py

@app.route('/')
def index_page(req):
    log.info(req)
    return {'Baka': 'Hello World!'}

@app.route('/home')
def home_page(req):
    log.info(req)
    return {'Route': 'home'}
```

1.2.3 App Folder

Untuk Struktur Application Folder optional

```
- root
  - package (AppBaka)
    - config ``optional, Baka(__name__, config_schema=True)``
      - config.yaml # digunakan for baka default configuration
    - __init__.py # the code goes in here
    - wsgi.py # for running in wsgi container e.g gunicorn
    - run.py # running development server
```

Default Configuration Baka from config.yaml

```
package: AppBaka # mandatory for root package
version: 0.1.0 # optional
baka:
  debug_all: True # mandatory for debug environment
  meta:
    version: 0.1.0 # mandatory for json response version
```

WSGI Container Application Server `wsgi.py`

```
# -*- coding: utf-8 -*-
"""
    WSGI Application Server
    ~~~~~

    :author: nanang.jobs@gmail.com
    :copyright: (c) 2017 by Nanang Suryadi.
    :license: BSD, see LICENSE for more details.

    wsgi.py
    """
from . import app

application = app
```

Running in Development mode `run.py`

```
# -*- coding: utf-8 -*-
"""

    ~~~~~

    :author: nanang.jobs@gmail.com
    :copyright: (c) 2017 by Nanang Suryadi.
    :license: BSD, see LICENSE for more details.

    run.py.py
    """
from . import app

app.run(use_reloader=True)
```

1.2.4 Running

Development mode

```
python run.py
```

Production mode with Gunicorn

```
gunicorn -w 1 -b 0.0.0.0:5000 AppBaka.wsgi
```

1.2.5 Contoh Aplikasi

```
git clone https://github.com/baka-framework/baka.git
```

```
cd examples  
  
python3 -m venv env  
  
source env/bin/active  
  
pip install baka  
  
python run.py
```


CHAPTER 2

Referensi API

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

Design notes, legal information and changelog are here for the interested.

3.1 Baka Changelog

Di sini Anda bisa melihat daftar lengkap perubahan di antara setiap rilis Baka framework.

3.1.1 Version 0.4

Major release, unreleased codename `aiko`

- add docs, [Bakadocs](#) untuk memudahkan cara menggunakan.
- Remove `app.scan()`, tidak perlu menggunakan fitur scan dari venusian.
- Dependency less venusian, `venusian.attach(wrapped, callback, 'pyramid', depth=depth)` menghilangkan dependency venusian di feature route.
- add `pyramid.authenticate` and `authorization` policy settings.
- add error handler function, `error_handler(code)` route, dengan memasukan http code untuk me-redirect ke handler yang sudah di buat.
- add exception handler function, `exception_handler` route, default code 509 dengan pesan status berisi error exception (*resiliency*) methodologi.
- add `trafaret` validator, default `False` untuk `config_schema` values.
- add directive `add_config_validator`, untuk schema `trafaret` validator.

- custom JSON renderer, `JSONEncoder`, type object dump (`ObjectId` untuk `mongodb` id); `uuid` type; `datetime` type; `EnumInt` Type; `enum`

3.2 License

Baka is licensed under a three clause BSD License. It basically means: do whatever you want with it as long as the copyright in Baka sticks around, the conditions are not modified and the disclaimer is present. Furthermore you must not use the names of the authors to promote derivatives of the software without written consent.

The full license text can be found below (*Baka License*). For the documentation and artwork different licenses apply.

3.2.1 Authors

Baka framework dibuat oleh Nanang Suryadi dan beberapa kontributor:

Development Lead

- Nanang Suryadi <nanang.jobs@gmail.com>

Patches and Suggestions

- *ini saat nya anda untuk berkontribusi*

3.2.2 General License Definitions

The following section contains the full license texts for Baka and the documentation.

- “AUTHORS” hereby refers to all the authors listed in the *Authors* section.
- The “*Baka License*” applies to all the source code shipped as part of Baka (Baka itself as well as the examples and the unittests) as well as documentation.

3.2.3 Baka License

Copyright (c) 2017 by Nanang Suryadi and contributors. See AUTHORS for more details.

Some rights reserved.

Redistribution and use in source and binary forms of the software as well as documentation, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- The names of the contributors may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE AND DOCUMENTATION IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE AND DOCUMENTATION, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

3.3 Cara berkontribusi di Baka framework

Terima kasih telah bersedia kontribusi Baka framework!

3.3.1 Pertanyaan Bantuan

Tolong, jangan menggunakan issue tracker untuk bertanya. Gunakan beberapa alternatif dibawah ini, untuk pertanyaan tentang kode anda:

- Email baka.framework@gmail.com untuk berdiskusi.
- Email nanang.jobs@gmail.com untuk berdiskusi langsung dengan creator seputar request feature dan konsultasi.

3.3.2 Pelaporan issues

- Jelaskan apa yang Anda harapkan terjadi.
- Jika memungkinkan, sertakan contoh [minimal, lengkap, dan dapat diverifikasi] untuk membantu kami mengidentifikasi masalah ini. Ini juga membantu memeriksa masalahnya tidak dengan kode anda sendiri
- Jelaskan apa yang sebenarnya terjadi. Termasuk traceback penuh jika ada adalah pengecualian
- Cantumkan versi Python, Baka, dan Pyramid Anda. Jika memungkinkan, periksa jika masalah ini sudah diperbaiki dalam repositori.