

WSGIの日々

なもなきフレームワーク

小田切篤 VISH株式会社

内容

- WSGIとは
- 使い方
- PythonPaste

WSGIとは

- Web Service Gateway Interface
- PEP333
- 統一インターフェイス

WSGIが必要なものは?

- 多くのフレームワーク
- 多くのアプリケーション
- サーバーに依存
- 依存をなくそう

WSGI ReadyなFW

- TurboGears
- Django
- Pylons
- web.py
- Trac
- Zope3

サーバー

- Apache (mod_wsgi, mod_fcgi, mod_scgi, ajp)
- Lighty (fcgi, scgi)
- Flup
- Paste
- Twisted Web2
- wsgiref

Hello, WSGI!

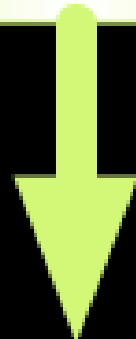
```
def application(environ, start_response):
    """Simple WSGI application"""
    status = '200 OK'
    response_headers = [('Content-type', 'text/plain')]
    start_response(status, response_headers)
    return ['Hello world!\n']

if __name__ == '__main__':
    httpd = make_server("", 8000, demo_app)
    httpd.serve_forever()
```

WSGIミドルウェア

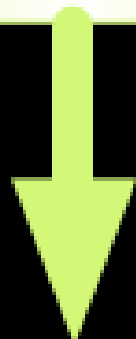
- デコレータのようなもの
- 認証、セッション、など
- 簡単に作れる

WSGIサーバー



`middle(environ, start_response)`

WSGIミドルウェア



`app(environ, start_response)`

WSGIアプリケーション

ミドルウェアの例

- wsgiauth 認証
- alchemyware SQLAlchemyセッション管理
- wsgiakismet SPAMブロッカー
- wsgiform 入力検証
- Deliverance HTMLオーバーレイ

フレームワーク

- RESTful
- 実行時自動生成
- Webセッションはサポートしない

RESTfull

- リソース主体
- リソースはURLを持つ
- リソースのURLは変わらない

実行時自動生成

- 作業を減らす
- 一貫性
- 妥当なデフォルト
- 上書きカスタマイズ

関連モジュール

- PythonPaste
- urlrelay
- Genshi
- ToscaWidgets
- FormEncode
- Twill

Paste

- ex. WSGIKit
- WSGI用ユーティリティセット
- Paste Deploy 設定関連
- Paste Script ユーティリティコマンド

urlrelay

- リクエストディスパッチャ
- デコレータ形式と登録形式

Flup

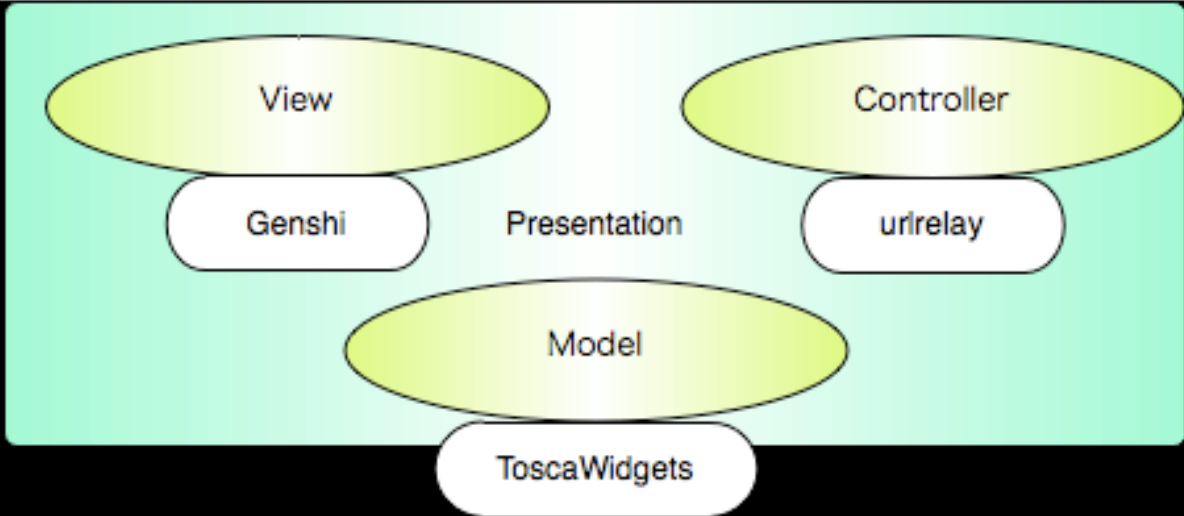
- ミドルウェア
- ajp, scgi, fcgiサービス
- PasteDeployから呼び出せる

アドレス帳

- 人を登録できる
 - 氏名
 - メールアドレス
- ユーザーごとに公開、非公開にできる
- ラベルを貼れる

アーキテクチャ

- MVC
 - Genshi
 - urlrelay
 - ToscaWidgets
- PAC
 - SQLAlchemy



URL

- Permalink = 更新URL
- 検索URL = ポストURL
- 編集フォーム = [検索URL]/edit,
[Permalink]/edit

Genshi

- XMLテンプレート
- XML, XPATHベース
- 継承はできません
- Pythonコードは書けません
- XIncludeでComposite

```
<div xmlns="http://www.w3.org/1999/xhtml"
      xmlns:py="http://genshi.edgewall.org/"
      xmlns:xi="http://www.w3.org/2001/XInclude"
      py:strip="">
  <dl>
    <dt>氏名</dt>

    <dd py:content="person.fullname">Full Name</dd>
    <dt>メールアドレス</dt>

    <dd><a py:content="person.email_address"
          py:attrs="dict(href='mailto:%s' %
person.email_address)">Mail Address</a>
    </dd>
  </dl>
</div>
```

WSGI構成

- urlrelay.URLReay リクエストディスパッチャ
- TWHostEngine ToscaWidget ミドルウェア
- Paste.Registry TWHostEngine に必要
- Paste.Auth 認証

[DEFAULT]

dburi = postgres://aodag:aodag@localhost/addressbook

[app:main]

paste.app_factory = addressbook:factory

with-filter = tw

[filter:tw]

paster.filter_app_factory = egg:ToscaWidgets#middleware

host_framework = wsgi

[server:main]

use = egg:PasteScript#wsgiutils

port = 8080

Twill でテスト

- WSGI対応
- シナリオによるテスト
- `py.test`と組み合わせれば保存と同時にテストできる!

現状

- GET系のURL生成できてます
- POST系のURL生成に着手しています

今後

- Quickstartできるように
- ATOM PPに対応
- ワークフローエンジンを取り込む

要するに

- いじくり倒せ！
- Paste, Twill は便利！
- Pythonicで行こう！