Python WSGI

@brett_langdon

http://brett.is

Web Server Gateway Interface

Specification for universal interface between web servers and web applications.

v1.0 PEP 333

http://legacy.python.org/dev/peps/pep-0333/

v1.0.1 PEP 3333

http://legacy.python.org/dev/peps/pep-3333/

Example

```python
from wsgiref.simple_server import make_server

def application(environ, start_response):
    start_response(
        '200 OK',
        [('Content-Type', 'text/plain')]
    )
    return ['Hello, WSGI\n']

if __name__ == '__main__':
    server = make_server('', 8000, application)
    server.serve_forever()
```

Example Explained

environ

http://legacy.python.org/dev/peps/pep-0333/#environ-variables

```
def application(environ, start_response):
    from pprint import pprint
    pprint(environ)
```

```
{'CONTENT_LENGTH': '',
 'CONTENT_TYPE': 'text/plain',
 'HTTP_ACCEPT': '*/*',
 'HTTP_HOST': 'localhost:8000',
 'HTTP_USER_AGENT': 'curl/7.30.0',
 'PATH_INFO': '/',
 'QUERY_STRING': '',
 'REMOTE_ADDR': '127.0.0.1',
 'REMOTE_HOST': '1.0.0.127.in-addr.arpa',
 'REQUEST_METHOD': 'GET',
 'SCRIPT_NAME': '',
 'SERVER_NAME': 'Bretts-MacBook-Pro.local',
 'SERVER_PORT': '8000',
 'SERVER_PROTOCOL': 'HTTP/1.1',
 'SERVER_SOFTWARE': 'WSGIServer/0.1 Python/2.7.5',
 'wsgi.errors': <open file '<stderr>' ...>,
 'wsgi.file_wrapper': <class wsgiref.util.FileWrapper..,
 'wsgi.input': <socket._fileobject ...>,
 'wsgi.multiprocess': False,
 'wsgi.multithread': True,
 'wsgi.run_once': False,
 'wsgi.url_scheme': 'http',
 'wsgi.version': (1, 0)}
```

Example Explained

start_response

http://legacy.python.org/dev/peps/pep-0333/#the-start-response-callable

```
start_response(status, response_headers, exc_info=None)
```

status

http://www.faqs.org/rfcs/rfc2616.html

```
"200 Ok"
```

response_headers

http://www.faqs.org/rfcs/rfc2616.html

```
[("Key", "Value"), ...]
```

Example Explained

return value

http://legacy.python.org/dev/peps/pep-0333/#buffering-and-streaming

```
def application(environ, start_response):
    return ['Hello', ', ', 'WSGI', '\n']
```

```
def application(environ, start_response):
    yield 'Hello, '
    yield 'WSGI\n'
```

Helpful Usages

URL Reconstruction

http://legacy.python.org/dev/peps/pep-3333/#url-reconstruction

```
from urllib import quote

def get_url(environ):
    url = environ['wsgi.url_scheme'] + '://'

    if environ.get('HTTP_HOST'):
        url += environ['HTTP_HOST']
    else:
        url += environ['SERVER_NAME']

        if environ['wsgi.url_scheme'] == 'https':
          if environ['SERVER_PORT'] != '443':
              url += ':' + environ['SERVER_PORT']
        else:
          if environ['SERVER_PORT'] != '80':
               url += ':' + environ['SERVER_PORT']

    url += quote(environ.get('SCRIPT_NAME', ''))
    url += quote(environ.get('PATH_INFO', ''))
    if environ.get('QUERY_STRING'):
        url += '?' + environ['QUERY_STRING']
    return url
```

Query String Dict

https://docs.python.org/2/library/urlparse.html#urlparse.parse_qs

```
form urlparse import parse_qs

def application(environ, start_response):
    query_string = parse_qs(environ['QUERY_STRING'])
    print query_string
    return []
```

Result

```
http://localhost:8000/?Key=Value
```

```
{'Key': ['Value']}
```

Reading POST Data

http://legacy.python.org/dev/peps/pep-3333/#environ-variables
http://webpython.codepoint.net/wsgi_request_parsing_post

```python
def application(environ, start_response):
    if environ['REQUEST_METHOD'] != 'POST':
        start_response('400 Bad Request', [
            ('Content-Type', 'text/plain'),
        ])
        return ['Only POST requests allowed']

    content_length = int(environ['CONTENT_LENGTH'])
    data = environ['wsgi.input'].read(content_length)
    start_response('200 Ok', [
        ('Content-Type', 'text/plain'),
    ])
    return [data]
```

Serve Static Files

http://legacy.python.org/dev/peps/pep-3333/#optional-platform-specific-file-handling

```
import os.path
ROOT_DIR = os.path.abspath('./public')

def application(environ, start_response):
    url = environ['PATH_INFO']
    if os.path.isfile(ROOT_DIR + url):
        start_response('200 Ok', [
            ('Content-Type', 'text/plain'),
        ])
        fp = open(ROOT_DIR + url, 'r')
        if 'wsgi.file_wrapper' in environ:
            return environ['wsgi.file_wrapper'](fp, 1024)
        else:
            return iter(lambda: fp.read(1024), '')
    else:
        start_response('404 Not Found', [
            ('Content-Type', 'text/plain'),
        ]);
        return ['Not Found']
```