**Cái đặt trong môi trường virtualenv (đang cài python 2.7):**

Pip install flask

**Tạo ứng dụng đầu tiên, tạo file helloflask.py với nd:**

From flask import Flask

from flask import Flask

app = Flask(\_\_name\_\_)

@app.route('/')

def hello\_world():

return 'Hello Flask'

**chạy dòng lệnh trên command line để khởi động :**

set FLASK\_APP=helloflask.py (tên file)

flask run (bật)

**Bật chế độ tự động reload lại khi code được sửa, hiển thị lỗi:**

set FLASK\_DEBUG=1

**Tạo các đường dẫn kèm theo biến truyền vào:**

@app.route('/user**/<username>**') (Biến để trong <> kiểu string mặc định)

def show\_user\_profile(username):

return 'User %s' % username

@app.route('/post/**<int:post\_id>**') (Chỉ rõ kiểu giá trị, int, float, string, path …)

def show\_post(post\_id):

return 'This is post %d' % post\_id

**Khác nhau giữa đường dẫn có gạch chéo và ko có:**

@app.route('/projects/') (như là 1 thư mục, khi vào chỉ vào url/projects sẽ tự động chuyển sang url/projects/)

def projects():

return 'This is project page'

@app.route('/about') (như là tên file – duy nhất nên khi vào url/about/ sẽ có báo lỗi – ko có thư mực nào là about)

def about():

return 'The about page'

**Sử dụng url\_for():**

>>> **from** flask **import** Flask**,** url\_for

>>> app = Flask**(**\_\_name\_\_**)**

>>> @app.route**(**'/'**)**

... **def** index**():** **pass**

...

>>> @app.route**(**'/login'**)**

... **def** login**():** **pass**

...

>>> @app.route**(**'/user/<username>'**)**

... **def** profile**(**username**):** **pass**

...

>>> **with** **app.test\_request\_context(): (cái này dùng để test)**

... print **url\_for(**'index'**)**

... print url\_for**(**'login'**)**

... print url\_for**(**'login'**,** next='/'**)**

... print url\_for**(**'profile'**,** username='John Doe'**)**

...

/

/login

/login?next=/

/user/John%20Doe

**HTTP methods:**

**from** flask **import** request

@app.route**(**'/login'**,** methods=**[**'GET'**,** 'POST'**])**

**def** login**():**

**if** request.method == 'POST'**:**

do\_the\_login**()**

**else:**

show\_the\_login\_form**()**

(mặc định là GET)

**Rendering Templates: ( vẽ giao diện), sử dụng hàm render\_template():**

**from** flask **import** render\_template

@app.route**(**'/hello/'**)**

@app.route**(**'/hello/<name>'**)**

**def** hello**(**name=**None):**

**return** render\_template**(**'hello.html'**,** name=name**)**

(những file html này nằm trong thư mục templates, còn những file static như css, js nằm trong thư mục static: static/style.css)

**File template có thể đuôi gì cũng được (txt, html, …) miễn tuân theo cú pháp:**

<!doctype html>

<title>Hello from admin</title>

{% if name %}

<h2>Hello {{ name }}</h2>

{% else %}

<h2>Hello, Flask here?</h2>

{% endif %}

</html>

**{% %} : dành cho câu lệnh**

**{{ }} : là biểu thức để in ra giao diện (print to template output)**

**{# #} : là comment**

**# : đây là dòng câu lệnh, tiện hơn {% %} nhưng 1 dòng có 2 cầu lệnh thì dùng {% %}**

**## đây là comment, tiện hơn {# #}**

Ví dụ:

# for item in seq:

<**li**>{{ item }}</**li**> ## this comment is ignored

# endfor

**Dòng lệnh có thể viết ra thành nhiều dòng nếu có dấu ngoặc đơn, ngoặc vuông hoặc ngoặc nhọn ? (** parentheses, braces or brackets) **:**

<**ul**>

# for href, caption in [('index.html', 'Index'),

('about.html', 'About')]:

<**li**><**a** href="{{ href }}">{{ caption }}</**a**></**li**>

# endfor

</**ul**>

**Truy cập tham số trong URL (?key=value), sử dụng thuộc tính args:**

searchword = request.args.get**(**'key'**,** ''**)**

**Cookies**

Truy cập cookies bằng thuộc tính cookies của đối tượng request:

request.cookies.get(‘param’)

**from** flask **import** request

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

**def** index**():**

username = request.cookies.get**(**'username'**)**

*# use cookies.get(key) instead of cookies[key] to not get a*

*# KeyError if the cookie is missing.*

Gán cookies bằng thuộc tính set\_cookie:

request.setcookie(‘username’, ‘the username’)

**from** flask **import** make\_response

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

**def** index**():**

resp = **make\_response(**render\_template**(**...**)) (chỉ rõ muốn làm gì)**

resp.set\_cookie**(**'username'**,** 'the username'**)**

**return** resp

Redirects và Errors : redirect() và abort()

**from** flask **import** abort**,** redirect**,** url\_for

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

**def** index**():**

**return** redirect**(**url\_for**(**'login'**))**

@app.route**(**'/login'**)**

**def** login**():**

abort**(**401**)**

this\_is\_never\_executed**()**

(chuyển hướng người dùng từ index sang trang login nhưng trang này họ ko được phép truy cập (401 : access denied)

**Bắt lỗi bằng hàm errorhandler:**

**from** flask **import** render\_template

@app.errorhandler**(**404**)**

**def** page\_not\_found**(**error**):**

**return** render\_template**(**'page\_not\_found.html'**),** 404

**Về Response:** giá trị trả về sẽ tự động chuyển thành response object, nếu giá trị trả về - return là string, sẽ dk chuyển thành response object với string đó là response body, 1 trạng thái status code 200 ok và kiểu text/html trong mimetype. Flask tự động chuyển kết quả trả về - return theo sau:

1. If a response object of the correct type is returned it’s directly returned from the view.
2. If it’s a string, a response object is created with that data and the default parameters.
3. If a tuple is returned the items in the tuple can provide extra information. Such tuples have to be in the form (response, status, headers) or (response, headers)where at least one item has to be in the tuple. The status value will override the status code and headers can be a list or dictionary of additional header values.
4. If none of that works, Flask will assume the return value is a valid WSGI application and convert that into a response object.

Muồn tùy chỉnh response theo ý mình: chỉ định dùng hàm make\_response()

@app.errorhandler**(**404**)**

**def** not\_found**(**error**):**

resp = make\_response**(**render\_template**(**'error.html'**),** 404**)**

resp.headers**[**'X-Something'**]** = 'A value'

**return** resp

**Logging:**

app.logger.debug**(**'A value for debugging'**)**

app.logger.warning**(**'A warning occurred (%d apples)'**,** 42**)**

app.logger.error**(**'An error occurred'**)**

**Sử dụng Flask Extensions:**

Link : <http://flask.pocoo.org/docs/0.12/extensions/#extensions>

**Triển khai lên web server:**

Link: <http://flask.pocoo.org/docs/0.12/deploying/#deployment>