

Introduction à Flask

Pyconfr 2011

Hobbestigrou

18 septembre 2011



- C'est un framework web
- un micro
- Il est initié par Armin Ronacher en 2010
- Il utilise les verbes http pour construire ces routes
- Une communauté assez active



- Comparaison rapide avec d'autres frameworks
- Zope est bien, mais lourd et complexe
- Django pris dans un modèle
- Pyramid, je ne connais pas



- Apporte quelque chose de nouveau
- Inspiré du monde Ruby par Sinatra
- Flask est léger, autonome
- Peu de dépendances
- Facile de prise en main
- Evolutif



Installation

```
$ sudo apt-get install python-flask
```

ou

```
$ mkdir myproject  
$ cd myproject  
$ virtualenv env  
$ . env/bin/activate  
$ sudo easy_install Flask
```



Exemple d'une application minimaliste

```
from flask import Flask

app = Flask(__name__)

@app.route('/')
def hello_world():
    return 'Salut pyconfr 2011'

if __name__ == '__main__':
    app.run()
```

- Embarque un serveur
- Affiche les requetes
- Ouvrez votre navigateur

```
python helloworld_flask.py
* Running on http://127.0.0.1:5000
```



Exemple d'une application minimaliste

- On commence par importer Flask
- On crée une instance
- On utilise un décorateur avec le chemin de l'url
- On utilise la fonction run pour lancer le serveur local avec notre application



- Par défaut le serveur est local
- En mode debug un utilisateur pour exécuter du code sur la machine
- Si le mode debug est désactivé, possibilité de rendre le serveur public

```
app.run(host='0.0.0.0')
```



- Reload le code
- Fournit un joli debugger en cas d'erreur

```
app.debug = True  
app.run()
```

ou

```
app.run(debug=True)
```

- Si vous utilisez un autre débbugger(PyDev), il faut désactiver le mode debug



- Les applications web moderne on de jolies url
- Une application flask est une collection de route
- Basiquement une route est un décorateur
- Liée à une méthode http
- Avec un path



- Flask est un aiguilleur, vers la bonne route
- Route trouvée, alors exécution du code



Les routes

- Chemins statiques

```
@app.route('/')
def index():
    return 'Index'

@app.route('/pyconfr')
def pyconfr():
    return 'Salut pyconfr 2011'
```

- Chemins avec tokens

```
@app.route('/user/<username>')
def show_user(username):
    # do something with username
    user = username.capitalize()
    return 'Salut' + ' ' + user
```

```
@app.route('/post/<int:post_id>')
def show_post(post_id)
    # show the post with the give
```



- int accepte des entiers
- float des flottants
- path accepte des slashes



- Par défaut c'est du GET
- Il est possible d'utiliser d'autres verbes HTTP

```
@app.route('/login', methodes=['GET', 'POST'])
def login():
    if request.method == 'POST':
        do_the_login
    else:
        show_the_login_form()
```



- GET retourne les informations stockées sur la page
- POST Envoie des informations au serveur
- DELETE Supprime l'information sur le serveur
- PUT Met à jour



- Rediriger sur une autre url

```
from flask import abort, redirect, url_for
app.route('/restricted')
def restricted()
    return redirect(url_for('/login'));
```



- Utilise Jinja2 par défaut
- Pour rendre un template, on utilise `render_template`



Moteur de template

```
from flask import render_template

@app.route('/hello/<name>')
def hello(name=None)
    return render_template('hello.html', name=name)
```

- Dans le cas d'un module

```
application.py
/templates
    /hello.html
```

- Dans le cas d'un package

```
/application
    /__init__.py
    /templates
        /hello.html
```



Flask
web development,
one drop at a time

```
from flask import Flask

app = Flask(__name__)

@app.route("/")
def hello():
    return 'Salut pyconfr 2011'

if __name__ == "__main__":
    app.run()
```

```
use Dancer;

get '/' => sub {
    return 'Salut pyconfr 2011';
};

dance;
```



Flask

web development,
one drop at a time

- Pas encore porter sur Python 3
- Werkzeug et Flask seront porté prochainement
- IL est recommandé d'utiliser Python 2.6 ou 2.7
- Des filtres
- C'est du wsgi



- [http ://flask.pocoo.org](http://flask.pocoo.org)
- Flaskr un moteur de blog écrit en Flask



- Flask est simple, efficace et succinct
- Aucune décision n'est prise pour nous
- Possibilités de faire un site en 2min
- Des filtres
- C'est du wsgi



- Des questions ?

