

DIPLOMADO DE PROGRAMACIÓN CON PYTHON V3

TEMA 7 FLASK

UNIBE

EDUCACIÓN
CONTINUA



Eliezer Figueroa
MCT, MOS, MCSA, MCSE, ITIL, SCF
3 Septiembre, 2019

COMPONENTES DEL TEMA

- ✓ FLASK FRAMEWORK
- ✓ ROUTE()
- ✓ VARIABLES
- ✓ URL_FOR()
- ✓ HTTP METHODS
- ✓ TEMPLATES



- Flask es un marco de aplicación web escrito en Python.
- Flask se basa en el kit de herramientas Werkzeug WSGI y el motor de plantillas Jinja2. Ambos son proyectos Pocco.
- Web Application Framework o simplemente Web Framework representa una colección de bibliotecas y módulos que permite a un desarrollador de aplicaciones web escribir aplicaciones sin tener que preocuparse por detalles de bajo nivel, como protocolos, administración de subprocesos, etc.



- Recomendamos usar la última versión de Python 3. Flask es compatible con Python 3.5 y versiones posteriores, Python 2.7 y PyPy.
- Puede usar un entorno virtual para administrar las dependencias del proyecto, tanto en desarrollo como en producción.
- Los entornos virtuales son grupos independientes de bibliotecas de Python, uno para cada proyecto. Los paquetes instalados para un proyecto no afectarán a otros proyectos ni a los paquetes del sistema operativo.

```
pip install Flask
```

- Recomendamos usar la última versión de Python 3. Flask es compatible con Python 3.5 y versiones posteriores, Python 2.7 y PyPy.
- Puede usar un entorno virtual para administrar las dependencias del proyecto, tanto en desarrollo como en producción.
- Los entornos virtuales son grupos independientes de bibliotecas de Python, uno para cada proyecto. Los paquetes instalados para un proyecto no afectarán a otros proyectos ni a los paquetes del sistema operativo.

```
pip install Flask
```

- La función **route()** de la clase **Flask** es un decorador, que indica a la aplicación qué URL debe llamar a la función asociada.

```
from flask import Flask
app = Flask(__name__)

@app.route('/')
def hello_world():
    return 'Hello World'

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

- Es posible crear una dirección URL dinámicamente, agregando partes variables al parámetro de regla.

```
from flask import Flask
app = Flask(__name__)

@app.route('/blog/<int:postID>')
def show_blog(postID):
    return 'Blog Number %d' % postID

@app.route('/rev/<float:revNo>')
def revision(revNo):
    return 'Revision Number %f' % revNo

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

- La función `url_for()` es muy útil para crear dinámicamente una URL para una función específica.

```
from flask import Flask, redirect, url_for
app = Flask(__name__)

@app.route('/admin')
def hello_admin():
    return 'Hello Admin'

@app.route('/guest/<guest>')
def hello_guest(guest):
    return 'Hello %s as Guest' % guest

@app.route('/user/<name>')
def hello_user(name):
    if name == 'admin':
        return redirect(url_for('hello_admin'))
    else:
        return redirect(url_for('hello_guest', guest = name))

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


- De forma predeterminada, la ruta Flask responde a las solicitudes GET.
- Sin embargo, esta preferencia se puede modificar proporcionando métodos específicos.

```
from flask import Flask, redirect, url_for, request
app = Flask(__name__)

@app.route('/success/<name>')
def success(name):
    return 'welcome %s' % name

@app.route('/login', methods = ['POST', 'GET'])
def login():
    if request.method == 'POST':
        user = request.form['nm']
        return redirect(url_for('success', name = user))
    else:
        user = request.args.get('nm')
        return redirect(url_for('success', name = user))

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

- En lugar de devolver HTML embebido en una función la función **render_template()** puede representar un archivo HTML.

```
from flask import Flask, render_template
app = Flask(__name__)

@app.route('/hello/<user>')
def hello_name(user):
    return render_template('hello.html', name = user)

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

```
<!doctype html>
<html>
  <body>
    <h1>Hello {{ name }}!</h1>
  </body>
</html>
```

- Tutorialspoint.com. (2019). Flask Tutorial. Retrieved 25 September, 2019, from <https://www.tutorialspoint.com/flask/index.htm>
- Palletsprojects.com. (2019). Templates. Retrieved 25 September, 2019, from <https://flask.palletsprojects.com/en/1.1.x/tutorial/templates/>

MUCHAS GRACIAS

Eliezer Figueroa
MCT,MOS,MCSA,MCSE,ITIL,SCF

UNIBE

EDUCACIÓN
CONTINUA

