

[TOC]

一、Flask 概述

1. 什么是Flask

1) Flask 介绍

Flask是一个基于Python并且依赖于Jinja2模板引擎和Werkzeug WSGI 服务的一个微型框架

WSGI：Web Server Gateway Interface(WEB服务网关接口)，定义了使用python编写的web app与web server之间接口格式

2) Flask 的框架模式 - MTV

1. 经典三层结构：MVC模式

- M：Models，模型层，负责数据库建模
- V：Views，视图层，用于处理用户显示的内容，如：html
- C：Controller，控制器，处理与用户交互的部分内容。处理用户的请求并给出响应

2. python常用：MTV模式

- M：Models，模型层，负责数据库建模
- T：Templates，模板层，用于处理用户显示的内容，如：html
- V：Views，视图层，处理与用户交互的部分内容。处理用户的请求并给出响应

2. 准备工作

1. 安装 flask

```
sudo pip3 install flask
```

2. 查看flask版本

- 进入python3交互模式：

```
>>>import flask
>>>flask.__version__
```

- 注意：不同版本之间会有细微差异，尽量以教学环境为主

3. 初始化flask应用

```
from flask import Flask
# 将当前运行得到主程序构建成Flask的应用，以便接收用户的请求(request),并给出响应(response)
app = Flask(__name__)
```

```
@app.route('/')
def index():
    return "<h1>this is my first flask app</h1>"

# 运行Flask应用
if __name__ == '__main__':
    app.run(debug=True)
```

- `@app.route()`: Flask中的路由定义, 定义用户的访问路径, / 表示的是整个网站的根路径
- `def index()`: 表示匹配上`@app.route()`路径后的处理程序-视图函数, 该类函数必须要有`return`, `return`后要给一个字符串 或 响应对象
- 运行应用后会启动flask自带的小型服务器, 默认在本机开启的端口号是5000
- `debug=True`, 是将当前的启动模式改为调试模式(开发环境中推荐使用调试模式, 生产环境中不允许使用)

二、Flask 使用

1. Flask-路由(route)

1) 什么是路由

- 客户端将请求发送给web服务器, web服务器再将请求发送给flask程序实例
- 程序实例需要知道每个url请求要运行哪些代码, 所以需要建立一个 url 到 python 函数的映射

路由就是处理url和python函数之间的关系的程序

在Flask中, 路由是通过 `@app.route` 装饰器来表示的

2) 路由的使用

1. 基本使用方式

```
@app.route('/')
def index():
    return 'This is index page.'

@app.route('/login')
def login():
    return 'This is login page.'
```

2. 带参数的路由

- 基本带参路由

```
@app.route('/show/<name>')
def show1(name):
```

```
# 在函数中 name 表示的就是地址栏上传递过来的数据
return 'xxx'
```

- 带多个参数的路由

```
@app.route('/show2/<name>/<age>')
def show1(name,age):
    # 在函数中 name 表示的就是地址栏上传递过来的数据
    return 'xxx'
```

- 指定参数类型的路由

```
@app.route('/show3/<name>/<int:age>')
def show1(name,age):
    # 在函数中 name 表示的就是地址栏上传递过来的数据
    return 'xxx'
```

3. 多 URL 的路由匹配

- 允许在一个视图处理函数中设置多个url路由规则

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

4. 路由中设置 HTTP 请求方法

- Flask路由规则也允许设置对应的请求方法，只有将匹配上请求方法的路径交给视图处理函数去执行
- 如果没有指定请求方法，默认允许GET请求

```
#只有post请求方式允许访问 localhost:5000/post
@app.route('/post',methods=['POST'])
def post():
    return 'xxxx'
```

3) 反向解析

- 正向解析：程序自动解析，根据@app.route()中的访问路径来匹配处理函数
- 反向解析：通过视图处理函数的名称自动生成视图处理函数的访问路径
 - Flask 中提供了 url_for() 函数，用于反向解析url

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

@app.route('/show/<name>')
def show(name):
    return "name:%s" % name
```

1. url_for('index'): 结果为 : /
2. url_for('show',name='qtx'): 结果为 : /show/qtx

2、Flask模板（templates）

1) 什么是模板

1. 模板是一个包含响应文本的文件(通常是html文件)
2. 模板中允许包含"占位变量"来表示动态的内容, "占位变量"最终会被真实的值所替换
3. 模板最终也会被解析成响应的字符串, 这一过程称为"渲染"

2) 模板的设置

1. 默认模板目录

默认情况下, Flask会在程序文件夹中的 `templates` 子文件夹中寻找模板

注意: 需要手动创建 `templates` 文件夹

2. 自定义模板文件的目录

可以修改配置, 为`template_folder`属性指定一个文件名字符串

```
app = Flask(__name__, template_folder='templates') # 配置模板文件的文件夹
```