Python语言基础与应用

基本扩展模块 / Web服务框架

陈斌 北京大学 gischen@pku.edu.cn

Web服务框架

- Web应用
- 框架的基本概念
- Flask框架
- 表单插件Flask-WTF

Web应用

- Web应用已经成为目前最热门的应用软件形式
- Web应用通过Web服务器提供服务，客户端采用浏览器或者遵循HTTP协议的客户端
- 由于需要处理HTTP传输协议，很多web开发框架涌现

框架的基本概念

什么是框架

Web服务器会处理与浏览器客户端交互的HTTP协议具体细节，但对具体内容的处理还需要自己编写代码

一个Web框架至少要具备处理浏览器客户端请求和服务端响应的能力

框架的基本概念

框架的特性

- 路由

  解析URL并找到对应的服务端文件或者Python服务器代码

- 模板

  把服务端数据合并成HTML页面。

- 认证和授权

  处理用户名、密码和权限

- Session

  处理用户在多次请求之间需要存储的数据

框架可能具备这些特性中的一种或多种

Flask框架

- Flask是一种非常容易上手的Python web开发框架，功能强大，支持很多专业Web开发需要的扩展功能

  Facebook认证和数据库集成

- 只需要具备基本的Python开发技能，就可以开发出一个web应用来

Flask框架：小例子

一个Web服务器测试

在浏览器中访问http://127.0.0.1:5000/，这个服务器会返回一行文本

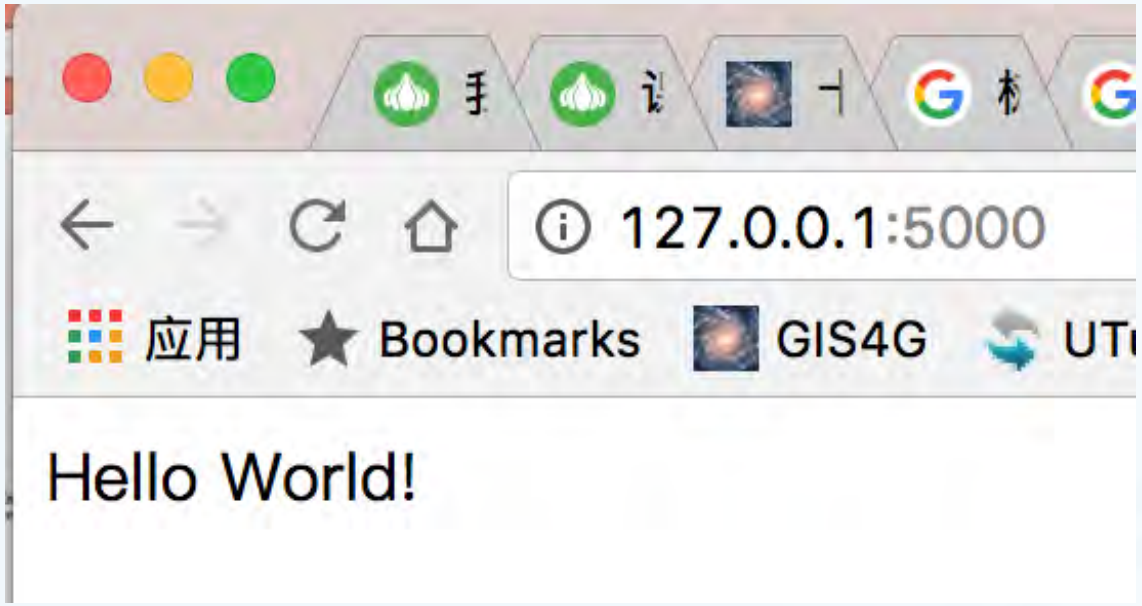
```
from flask import Flask
app = Flask(__name__)

@app.route("/")
def hello():
    return "Hello World!"

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

Flask框架：小例子

```
============= RESTART: /Users/chenbin/Documents/homework/flsk.py ===========
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [31/Mar/2017 02:47:59] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [31/Mar/2017 02:48:00] "GET /favicon.ico HTTP/1.1" 404 -
```



表单插件Flask-WTF

关于表单的扩展库

使用Flask-WTF时，每个表单都抽象成一个类

```
from flask_wtf import Form
from wtforms import StringField
from wtforms.validators import DataRequired

class MyForm(Form):
    user = StringField('Username', validators=[DataRequired()])

from flask import Flask, render_template

app = Flask(__name__)
app.secret_key = '1234567'

@app.route('/login', methods=('GET', 'POST'))
def login():
    form = MyForm()
    if form.validate_on_submit():
        # if form.user.data == 'admin':
        if form.data['user'] == 'admin':
            return 'Admin login successfully!'
        else:
            return 'Wrong user!'
    return render_template('login.html', form=form)

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

表单插件Flask-WTF

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>登录表单</title>
</head>
<body>
<form method="POST" action="{{ url_for('login') }}">
    {{ form.hidden_tag() }}
    {{ form.user.label }}: {{ form.user(size=20) }}
    <input type="submit" value="Submit">
</form>
</body>
</html>
```