

Flask



第八小组

20121034 胡才郁

20121036 黄逸弘

20121054 徐鼎力

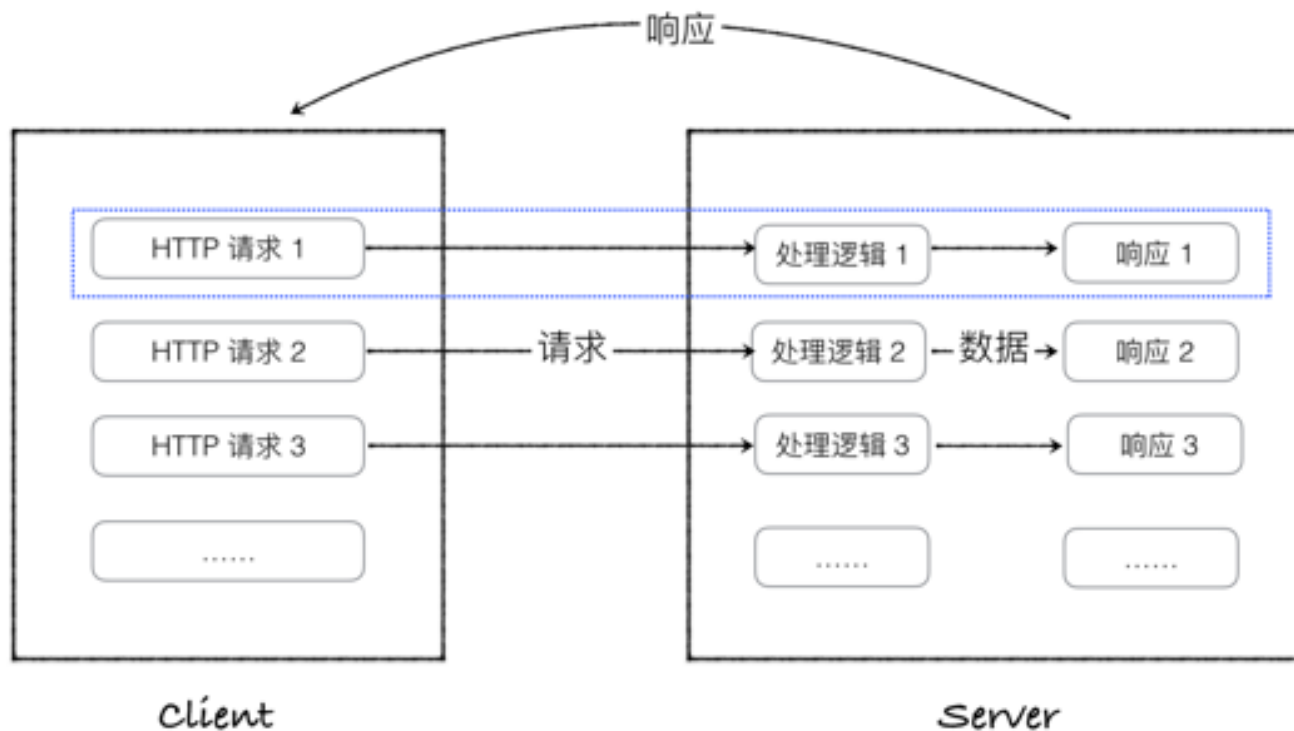
20121059 朱若时

20121706 张俊雄

Web框架



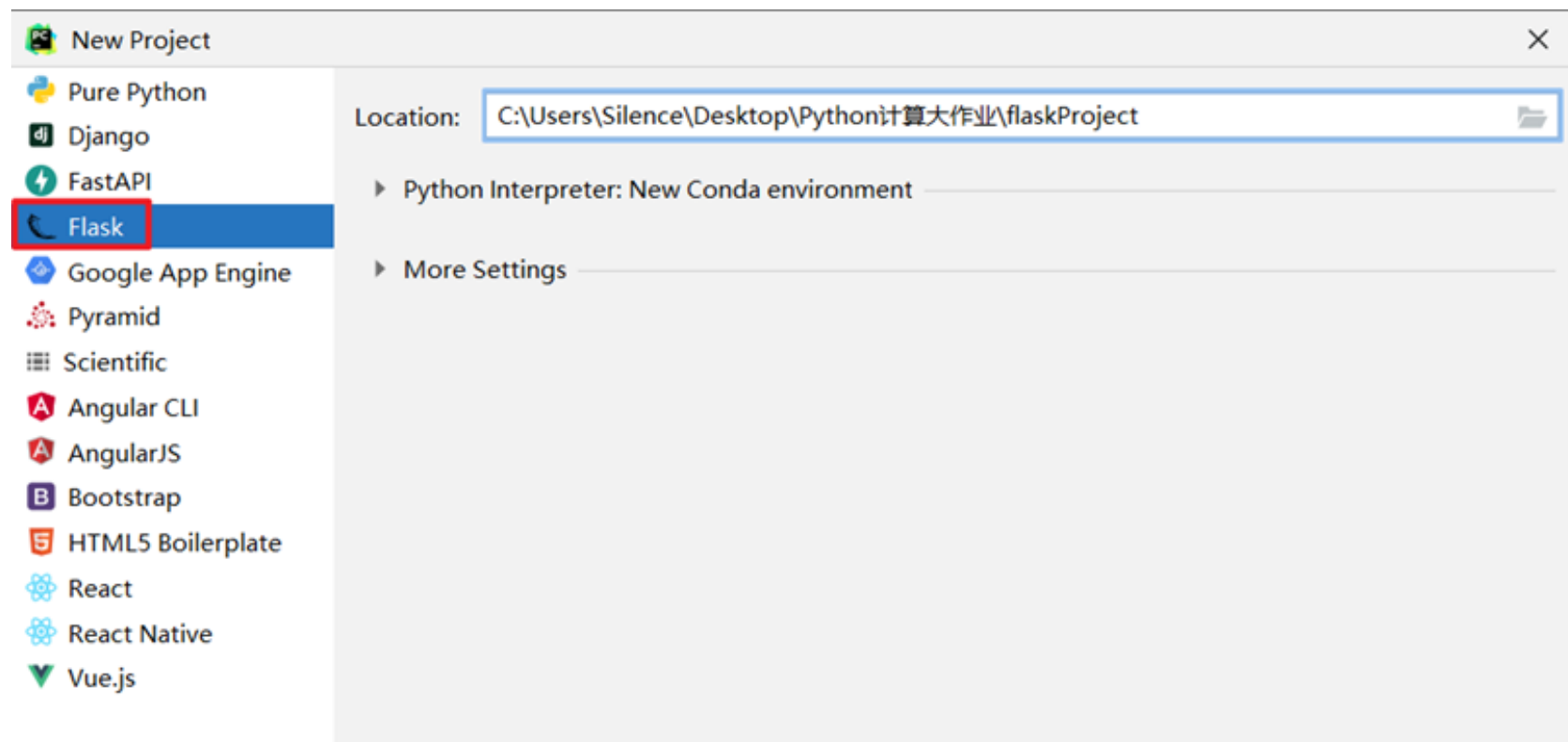
- 所有Flask程序都必须创建一个程序实例。
- 当客户端想要获取资源时，一般会通过浏览器发起HTTP请求。
- Web服务器把来自客户端的请求都交给Flask程序实例。
- Flask使用来做路由分发在Flask程序中，路由一般是通程序实例Python的装饰器实现。
- 通过调用视图函数，获取到数据后，然后由Flask返回响应数据给浏览器，最后浏览器显示返回的结果。



PyCharm

对于主流

框架支持



简单启动



代码实现

```
# 导入Flask类
from flask import Flask

# Flask类接收一个参数__name__
app = Flask(__name__)

# 装饰器的作用是将路由映射到视图函数index
@app.route('/')
def index():
    return 'Hello World'

# Flask应用程序实例的run方法启动WEB服务器
if __name__ == '__main__':
    app.run()
```

浏览器访问



Hello World

路由传参



代码实现

```
# 路由传递的参数默认当做string处理, 这里指定int, 尖括号中冒号后面的内容是动态的
@app.route('/user/<int:id>')
def router_test(id):
    return '%d' % id
```

浏览器访问



服务器输出

```
* Serving Flask app 'app.py' (lazy loading)
* Environment: development
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [23/May/2022 18:17:34] "GET /user/20121034 HTTP/1.1" 200 -
```

数据库配置



```
from flask import Flask
from flask_sqlalchemy import SQLAlchemy

app = Flask(__name__)

#设置连接数据库的URL
app.config['SQLALCHEMY_DATABASE_URI'] = 'mysql://root:root@127.0.0.1:3306/Flask_test'

#设置每次请求结束后会自动提交数据库中的改动
app.config['SQLALCHEMY_COMMIT_ON_TEARDOWN'] = True

app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = True
#查询时会显示原始SQL语句
app.config['SQLALCHEMY_ECHO'] = True
db = SQLAlchemy(app)
```



ORM

ORM是通过使用描述对象和数据库之间映射的元数据，将程序中的对象自动持久化到关系数据库中。

例如:查找用户表中id为2的学生

说人话：不写sql语句，直接操作对象

不使用ORM的Sql写法

```
sql = "SELECT ... FROM persons WHERE id = 2"
```

User表

	id	name	email	pswd	role_id
1	1	wang	wang@163.com	123456	1
2	2	zhang	zhang@189.com	201512	2
3	3	chen	chen@126.com	987654	2
4	4	zhou	zhou@163.com	456789	1

使用ORM的写法

```
User.query.filter(User.id = 2).all()
```

数据库支持

Flask-SQLAlchemy



常用的SQLAlchemy字段类型

类型名	python中类型	说明
Integer	int	普通整数，一般是32位
SmallInteger	int	取值范围小的整数，一般是16位
BigInteger	int或long	不限制精度的整数
Float	float	浮点数
Numeric	decimal.Decimal	普通整数，一般是32位
String	str	变长字符串
Text	str	变长字符串，对较长或不限长度的字符串做了优化
Unicode	unicode	变长Unicode字符串
UnicodeText	unicode	变长Unicode字符串，对较长或不限长度的字符串做了优化
Boolean	bool	布尔值
Date	datetime.date	时间
Time	datetime.datetime	日期和时间
LargeBinary	str	二进制文件

数据库支持

Flask-SQLAlchemy



常用的SQLAlchemy列选项

选项名	说明
primary_key	如果为True，代表表的主键
unique	如果为True，代表这列不允许出现重复的值
index	如果为True，为这列创建索引，提高查询效率
nullable	如果为True，允许有空值，如果为False，不允许有空值
default	为这列定义默认值

常用的SQLAlchemy关系选项

选项名	说明
backref	在关系的另一模型中添加反向引用
primary join	明确指定两个模型之间使用的联结条件
uselist	如果为False，不使用列表，而使用标量值
order_by	指定关系中记录的排序方式
secondary	指定多对多中记录的排序方式
secondary join	在SQLAlchemy中无法自行决定时，指定多对多关系中的二级联结条件

模型类配置



```
class Role(db.Model):
    # 定义表名
    __tablename__ = 'roles'
    # 定义列对象
    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(64), unique=True)
    us = db.relationship('User', backref='role')

    #repr()方法显示一个可读字符串
    def __repr__(self):
        return 'Role:%s'% self.name

class User(db.Model):
    __tablename__ = 'users'
    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(64), unique=True, index=True)
    email = db.Column(db.String(64), unique=True)
    pswd = db.Column(db.String(64))
    role_id = db.Column(db.Integer, db.ForeignKey('roles.id'))

    def __repr__(self):
        return 'User:%s'%self.name
```

Role表

	id	name
1	1	admin
2	2	user

User表

	id	name	email	pswd	role_id
1	1	wang	wang@163.com	123456	1
2	2	zhang	zhang@189.com	201512	2
3	3	chen	chen@126.com	987654	2
4	4	zhou	zhou@163.com	456789	1

增

```
# 创建表
db.create_all()
# 删除表
db.drop_all()

# 插入数据
ro1 = Role(name='admin')
db.session.add(ro1)
db.session.commit()
# 再次插入一条数据
ro2 = Role(name='user')
db.session.add(ro2)
db.session.commit()
# 一次插入多条数据
us1 = User(name='wang', email='wang@163.com', pswd='123456', role_id=ro1.id)
us2 = User(name='zhang', email='zhang@189.com', pswd='201512', role_id=ro2.id)
us3 = User(name='chen', email='chen@126.com', pswd='987654', role_id=ro2.id)
us4 = User(name='zhou', email='zhou@163.com', pswd='456789', role_id=ro1.id)
db.session.add_all([us1, us2, us3, us4])
db.session.commit()
```

删



```
# 查询数据后删除
user = User.query.first()
db.session.delete(user)
db.session.commit()
User.query.all()
```

查

```
# 模糊查询
User.query.filter_by(name='wang').all()
# 返回名字结尾字符为g的所有数据
User.query.filter(User.name.endswith('g')).all()
# 主键查询
User.query.get()
# 逻辑与
from sqlalchemy import and_
User.query.filter(and_(User.name≠'wang',User.email.endswith('163.com'))).all()
# 逻辑或
from sqlalchemy import or_
User.query.filter(or_(User.name≠'wang',User.email.endswith('163.com'))).all()
```

改



```
# 更新数据
user = User.query.first()
user.name = 'dong'
db.session.commit()
User.query.first()
```

夹带一些私货(tù cǎo)...



Flask作为Web框架

个人认为的适用范围?

- 社区小、资料难查
- 工作岗位少、需求小
- 生态不好、工具少、重复造轮子
- 复杂的数据接口难以编写
- Python作为第一语言的同学
- Python计算大作业....数据库大作业.... 各种大作业....
- 编写不需要定制的数据接口
- 机器学习项目WEB端展示

Flask与其他Web框架播放量对比

springboot

影视 0

直播 0

专栏 99+

话题 0

用户 6

最多弹幕

最多收藏

钟

30-60分钟

60分钟以上

音乐

舞蹈

游戏

知识

科技

运动

汽车

生活

SpringBoot2

200万播放真干货

224.9万 2.5万 26:12:01

尚硅谷雷神SpringBoot2零基础入门springboot全套完整版 (spring boot)

UP 尚硅谷 · 2020-12-22

全面透析SpringBoot

338.5万 3.1万 16:46:15

最强SpringBoot实战教程内容通俗易懂springboot微服务开发实战SpringBoot

UP 女毒v · 2020-10-26

flask

影视 0

直播 0

专栏 99+

话题 0

用户 3

最多弹幕

最多收藏

钟

30-60分钟

60分钟以上

音乐

舞蹈

游戏

知识

科技

运动

汽车

生活

5个月后的你会感谢现在的自己

27.3万 1680 04:46:48

黑马程序员-6节课入门Flask框架web开发视频

UP 白嫖无双 · 2018-2-20

Flask

16.2万 1283 88:19:01

【千锋】零基础玩转Python_Flask (flask快速上手)

UP 千锋python · 2020-6-26

夹带一些私货(tù cǎo)...



Flask作为Web框架

- 社区小、资料难查
- 工作岗位少、需求小
- 生态不好、工具少、重复造轮子
- 复杂的数据接口难以编写

个人认为的Flask适用范围?

- Python作为第一语言的同学
- Python计算大作业....数据库大作业.... 各种大作业....
- 编写不需要定制的数据接口
- 机器学习项目WEB端展示

夹带一些私货(tù cǎo)...



- 复杂的数据接口难以编写

如果用Flask实现，痛苦！

```
{
  "code": 200,
  "message": "商户信息查询成功",
  "data": {
    "vendorName": "康师傅工厂",
    "vendorImg": "https://img.zcool.cn/community/0193bc5de4878fa80120686b4f4c9c.jpg@1280w_1l_2o_100sh.jpg",
    "totalSold": 1,
    "averagePrice": 100,
    "announcement": "好吃",
    "inBusiness": 1,
    "categoryInfoVoList": [
      {
        "vendorId": null,
        "categoryName": "猪肉脯",
        "productVendorVoList": [
          {
            "productName": "加力加虾条5",
            "productId": 23,
            "productImg": "https://cbu01.alicdn.com/img/ibank/2014/661/823/1719328166_1541695327.jpg",
            "sellPrice": 100,
            "soldNum": 174,
            "stock": 94,
            "productStatus": 1
          }
        ]
      }
    ]
  },
}
```


夹带一些私货(tù cǎo)...



Flask作为Web框架

- 社区小、资料难查
- 工作岗位少、需求小
- 生态不好、工具少、重复造轮子
- 复杂的数据接口难以编写

个人认为的Flask适用范围?

- Python作为第一语言的同学
- Python计算大作业....数据库大作业.... 各种大作业....
- 编写不需要定制的数据接口
- 机器学习项目WEB端展示

谢谢观看

FLASK



20121034 胡才郁