

Python API 接口开发用法介绍

API（Application Programming Interface）是应用程序接口的简称

```
# 定义一个API接口
@app.route("/")
def index():
    return "<a href='/list'><img src='/static/py11.png'></a>"

# 定义一个API接口
@app.route("/addSubmit", methods=["POST"])
def addSubmit():
    tid = request.form.get("tid")
    tname = request.form.get("tname")
    tcontent=request.form.get("tcontent")
    leixing=MovieType(tid,tname,tcontent)
    result=mtdal.insert(leixing)
    if result>0:
        return "插入成功 <a href='/list'>刷新</a>"
    else:
        return "插入失败 <a href='/list'>刷新</a>"
```

Pandas

Pandas 是 Python 语言的一个扩展程序库，用于数据分析。

Pandas 名字衍生自术语 "panel data"（面板数据）

Pandas 可以从各种文件格式比如 CSV、JSON、Excel

python 终端输入安装：

```
pip3 install -i https://pypi.tuna.tsinghua.edu.cn/simple pandas
```

Pandas 数据结构 DataFrame

```
data = {"Site":["Google", "Runoob", "Wiki"], "Age":[10, 12, 13], "sss":
[22, 33, 44]}
df = pd.DataFrame(data)
print(df.loc[1]) #一行
print(df.loc[[0,1]]) # 多行
print(df["Age"]) # 一列
print(df[["Age", "Site"]]) # 多列
```

```

print(df["Age"][1]) # 一个值
print(df[(df.Age>11) & (df.sss>35)])# 带条件筛选
print(df[(df.Age>11) & (df.sss>35)]["Age"])
print(df[(df.Age.astype(int)>11) & (df.sss>35)]) #如果需要转换数据类型

#修改值-查出来后，右侧给左侧赋值即可，不存在的列即为添加
df["Age"]=100
df["Age"][1]=99
    #添加列
df["tid"]=2
df["Score"]=[90,40,99]
df.insert(1,"Sex",["男","女","男"])
    # 1代表列 0代表行
df.drop(1,axis=0,inplace=True)
df.drop([0,1],axis=0,inplace=True)
df.drop("Age",axis=1,inplace=True)
df.drop(df[df.Age>10].index,axis=0,inplace=True)

```

Pandas 处理 CSV 文件

```

df = pd.read_csv("/root/shell/douban.csv")
df = df[["id", "title", "rate"]]
df.to_csv("/root/python/db.csv",index=False)
df.to_csv("/root/python/db.csv",mode="a", header=False, index=False) #追加

```

Pandas 处理 JSON

```

import pandas as pd
d2 = [
{"class": "Year 1", "student number": 20, "room": "Yellow"},
{"class": "Year 2", "student number": 25, "room": "Blue"} ]
df = pd.DataFrame(d2)
print(df)

```

douban.txt 的文件处理

```

#内嵌的方法
import pandas as pd
import json
dict1=None
with open("douban.txt",mode="r",encoding="UTF-8") as f:
    dict1 = json.loads(
        f.read()) #可以从文件读取过来

```

```
f.close()
df = pd.DataFrame(dict1["subjects"])
df=df[["id", "rate"]] #改变顺序和列数
print(df)
df.to_csv("doubanout1.csv", index=False)
```

Pandas 处理 excel 文件

sheet_name 指定了读取 excel 里面的哪一个 sheet

usecols 指定了读取哪些列

nrows 指定了总共读取多少行

header 指定了列名在第几行，并且只读取这一行往下的数据

index_col 指定了 index 在第几列

engine="openpyxl" 指定了使用什么引擎来读取 excel 文件

安装命令：

```
import pandas as pd
df = pd.read_excel("student.xlsx", sheet_name="Sheet1", header=1)
print(df)
df.to_excel("student1.xlsx", sheet_name="xs", index=False)
```

爬虫

安装

```
pip3 install -i https://pypi.tuna.tsinghua.edu.cn/simple requests
pip3 install urllib3==1.26.15
```

爬取一条数据

数据包的 headers 里面有我们需要的所有数据

数据包的 response 里面有我们要传递的 json 数据

```
headers={"User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.71 Safari/537.36"}
url="https://movie.douban.com/j/chart/top_list"
params=
{"type": "25", "interval_id": "100:90", "action": "", "start": "0", "limit": "1"}
ck={"Cookie": '自己浏览器的'}
response=requests.get(url=url, headers=headers, params=params, cookies=ck)
print(response.json())
```

爬取所有数据

```
import requests
import pandas as pd
url1="https://movie.douban.com/j/chart/top_list"
params1={"type":20,"interval_id":"100:90","action":"","start":0,"limit":400}
header1={"user-agent":"Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/137.0.0.0 Safari/537.36
Edg/137.0.0.0"}
res=requests.get(url=url1,params=params1,headers=header1)
content=res.json()
df=pd.DataFrame(content)
print(df)
```

pandas 分析处理数据

```
content=response.json()
df=pd.DataFrame(content)
df=df[["id","title","release_date","score"]]
df["tid"]=1
df.to_csv("/root/python/movie.csv",mode="a",header=False,index=False)

# 去掉重复数据 去掉后面的
df=pd.read_csv("/root/python/movie.csv")
df=df.drop_duplicates(subset=['id','title','release_date','score'])
df.to_csv("/root/python/movie1.csv",index=False)

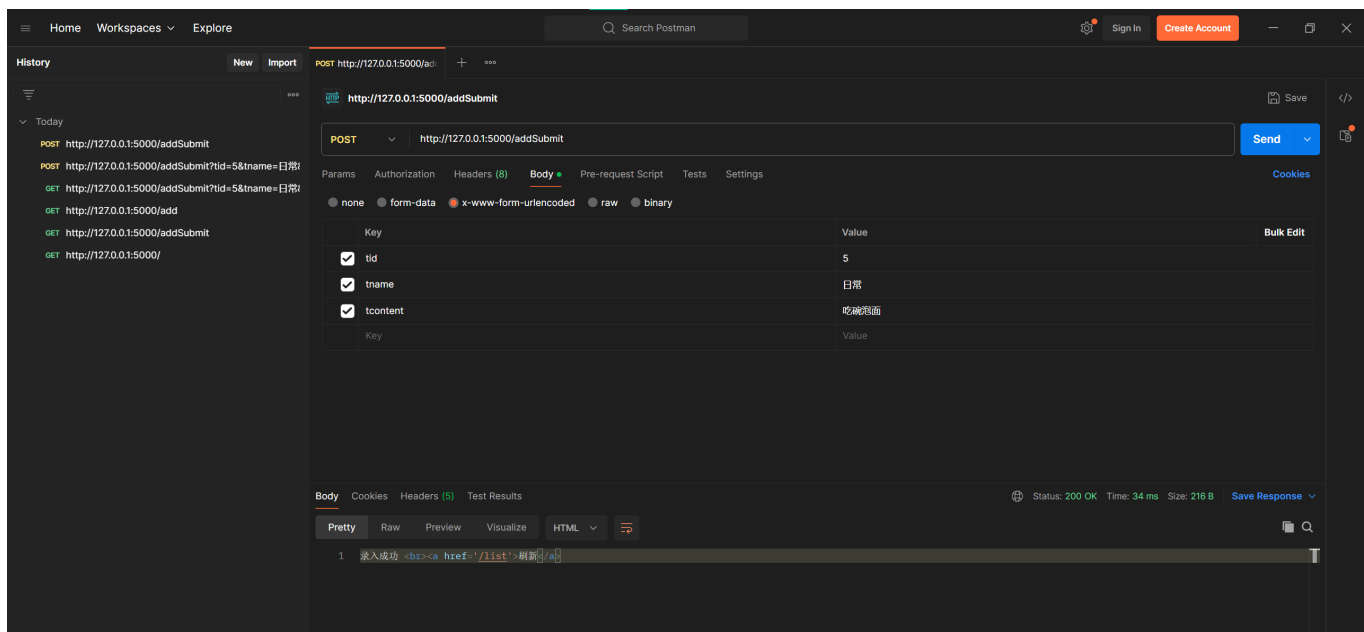
# 修改mysqlcsv.sh 导入到mysql数据库
os.system("cp /root/python/movie1.csv /usr/local/mysql/data/movie1.csv")

os.system("/root/shell/mysqlcsv.sh Movie /usr/local/mysql/data/movie1.csv")
```

作业练习

1.安装 postman ， 测试添加电影类型接口是否正确

- 地址栏<http://127.0.0.1:5000/addSubmit>
- 设置POST传递方式
- 设置Body中的x-www-form-urlencoded
- 填写数据后点击send提交



上机练习14

2.创建以日期时间为名字的日志文件，格式如 20230303102030.log

```
import datetime
import os
```

```
current_time = datetime.datetime.now()
filename = current_time.strftime("%Y%m%d%H%M%S.log")
os.system(f"touch /root/python/{filename}")
```

3.列表 `[["Google",10],["Runoob",12],["Wiki",13]]`，转成 DataFrame 数据打印

```
import pandas as pd

data=[["Google",10],["Runoob",12],["Wiki",13]]
df=pd.DataFrame(data)
print(df)
```

4.将 `[{"a": 1, "b": 2}, {"a": 5, "b": 10, "c": 20}]` 转成 DataFrame 数据，提取第 2 行 a 那一列的数据

```
import pandas as pd

data=[{"a": 1, "b": 2}, {"a": 5, "b": 10, "c": 20}]
df=pd.DataFrame(data)
print(df.loc[1, "a"])
```

5.将 `data = {"chinese": [89,97,68,56,88,77], "math": [99,67,100,78,89,66], "english": [73,57,89,90,82,55]}` 转成 DataFrame 数据，提取三科分数都及格的数据

```
import pandas as pd
data = {"chinese": [89,97,68,56,88,77],
```

```

        "math": [99, 67, 100, 78, 89, 66],
        "english": [73, 57, 89, 90, 82, 55]
    }
df=pd.DataFrame(data)
print(df[(df.chinese>60) & (df.math>60) & (df.english>60)])

# 6.使用 pandas 处理 douban.txt，提取列 id,title,rate，并且提取 rate 大于 7.5 的行
导出 douban1.csv
import pandas as pd
import json
dict1=None
with open("/root/python/douban.txt", mode="r", encoding="UTF-8") as f:
    dict1 = json.loads(f.read())

df=pd.DataFrame(dict1["subjects"])
df=df[df.rate.astype(float)>7.5][["id", "title", "rate"]]
df.to_csv("/root/python/douban1.csv", index=False)

# 7.把 order.xlsx 使用 mobox 传入到 /root/python 中，根据文档做如下操作
# 把文档数据（从列名开始）转换为 dataframe 输出
import pandas as pd
df=pd.read_excel("/root/python/order.xlsx", sheet_name="Sheet1", header=1)
print(df)
# 在产品后面增加一列采购人，内容为 ["坤坤", "杰杰", "坤坤", "丽丽", "丽丽", "坤坤"]
df.insert(2, "采购人", ["坤坤", "杰杰", "坤坤", "丽丽", "丽丽", "坤坤"])

# 查看金额大于 100 的坤坤的订单列表
print(df[(df.金额>100) & (df.采购人=="坤坤")])

# 查看金额大于 100 的坤坤的订单列表的产品及金额列
df=df[(df.金额>100) & (df.采购人=="坤坤")][["产品", "金额"]]
print(df)

# 把上一步的结果另存为 order_kunkun.xlsx，表单名称为坤坤，不显示索引，如下：
# 产品 金额
# 投影仪 2000
# 打印机 298
df.to_excel("/root/python/order_kunkun.xlsx", sheet_name="坤坤", index=False)

```

```

# 上机练习15
# 爬取至少五种电影类型（动画、科幻、）的电影追加到 csv 文件中
# 每次更换类型进行爬取需要修改params1的"type":20中的数字（豆瓣网的），以及df["tid"]=6的
数字（自己网站对应类型的编号）
import requests
import pandas as pd

```

```

import os
url1="https://movie.douban.com/j/chart/top_list"
params1={"type":20,"interval_id":"100:90","action":"","start":0,"limit":400}
header1={"user-agent":"Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/137.0.0.0 Safari/537.36
Edg/137.0.0.0"}
res=requests.get(url=url1,params=params1,headers=header1)
content=res.json()
df=pd.DataFrame(content)
df=df[["id","title","release_date","score"]]
df["tid"]=6
# df.to_csv("/root/python/getdouban.csv",index=False) # 第一次执行,覆盖写
df.to_csv("/root/python/getdouban.csv",index=False,mode="a",header=False) #添加数据

# 把csv里面重复的数据去掉, 导入到 mysql 数据库的 Movie 表中
import pandas as pd
import os
df=pd.read_csv("/root/python/getdouban.csv")
df=df.drop_duplicates(subset=["id","title","release_date","score"])
df.to_csv("/root/python/getdouban.csv",index=False)
os.system("cp /root/python/getdouban.csv /usr/local/mysql/data/getdouban.csv")
os.system("/root/python/mysqlcsv.sh Movie
/usr/local/mysql/data/getdouban.csv")

# mysqlcsv.sh内容:

#设置mysql连接参数
dbhost="127.0.0.1" #主机名
dbprot=3306 #端口
dbuser="root" #用户名
dbpass="root123456" #密码
db="test" #表空间
#导入数据
sql3="""LOAD DATA INFILE '$2' INTO TABLE $1
CHARACTER SET utf8
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 LINES"""
mysql -h$dbhost -P$dbprot -u$dbuser -p$dbpass $db -e "$sql3"
echo "导入完成"

```

其他Python库

1. NumPy (Numerical Python)

- **核心功能：**提供多维数组对象（`ndarray`）、高效的数学函数、线性代数运算和随机数生成。
- **应用场景：**科学计算、数据分析、机器学习（如 TensorFlow 依赖 NumPy 数组）。
- **优势：**底层用 C 实现，运算速度极快，是众多科学库的基础。 **示例：**

```
import numpy as np
a = np.array([1, 2, 3]) # 创建一维数组
print(a.mean()) # 计算平均值
```

2. Matplotlib

- **核心功能：**2D 绘图库，支持线图、散点图、柱状图、饼图等多种可视化类型。
- **应用场景：**数据可视化、论文图表、交互式绘图。
- **搭配工具：**常与 pandas（数据处理）和 seaborn（高级可视化）结合使用。
- **示例：**

```
import matplotlib.pyplot as plt
x = [1, 2, 3]
y = [4, 5, 6]
plt.plot(x, y) # 绘制线图
plt.show()
```

3. unittest

- **核心功能：**Python 内置的单元测试框架，支持测试用例、测试套件、断言等。
- **应用场景：**软件开发中的测试驱动开发（TDD）、代码质量保障。
- **对比工具：**类似 pytest，但 unittest 更适合初学者和小型项目。
- **示例：**

```
import unittest

def add(a, b):
    return a + b

class TestAdd(unittest.TestCase):
    def test_add(self):
        self.assertEqual(add(1, 2), 3)

if __name__ == '__main__':
    unittest.main()
```

4. Setuptools

- **核心功能：**Python 包开发工具，用于打包、分发和安装 Python 项目。
- **关键文件：** `setup.py`（定义包元信息）、`requirements.txt`（依赖列表）。
- **应用场景：**发布开源库到 PyPI（Python Package Index）或内部使用。
- **示例：**

```
# setup.py
from setuptools import setup, find_package

setup(
    name='mypackage',
    version='1.0.0',
    packages=find_packages(),
)
```

5. Jupyter Notebook

- **核心功能：**交互式计算环境，支持代码、Markdown 文本、公式和可视化混排。
- **应用场景：**数据分析、机器学习实验、学术论文写作（如 Jupyter Lab）。
- **优势：**实时运行代码，便于分享和演示，支持多种编程语言（通过内核）。
- **启动命令：**

```
jupyter notebook # 启动传统界面
jupyter lab      # 启动更现代化的Jupyter Lab
```

总结

库名	主要用途	典型场景
NumPy	高性能数组计算	科学计算、AI 框架基础
Matplotlib	数据可视化	论文图表、交互式绘图
unittest	单元测试	代码质量保障
Setuptools	包打包与分发	发布 Python 库到 PyPI
Jupyter	交互式计算与文档	数据分析、学术研究