

周瑞发的网站

欢迎访问



4 min. read

Python程序设计作业#7

📅 2021-07-16 | 📅 2020-12-08 | 👁 2 | 💬 0

Python程序设计#7作业

截止时间：2020年12月7日23:59:59

作业题目

实现remoteProxy的用户数据库的REST管理接口

基于Sanic实现对user.db数据库（SQLite）的管理接口

支持对user用户的增、删、改、查操作。

初步掌握Sanic的使用方法。

remoteRest代码

remoteRest代码嵌入下方的code block中。

```
1  from sanic import Sanic
2  from sanic import response
3  from sanic import exceptions
4  from sanic.response import json
5  import aiosqlite
6  import logging
7  import signal
8  import json
9
```

```

10 app = Sanic("RemoteProxyAdmin")
11 app.config.DB_NAME = 'user.db'
12 @app.exception(exceptions.NotFound)
13 async def ignore_404(req, exc):
↑ 14     return response.text('errUrl', status=404)
15
16 @app.get('/user')
17 async def userList(req):
18     userList = list()
19     async with aiosqlite.connect(app.config.DB_NAME) as db:
20         async with db.execute("SELECT name, password, dataRate FROM user;") as cu
21             async for row in cursor:
22                 user = {'name' : row[0], 'password' : row[1], 'dataRate:' : row[2]
23                 logging.debug(f'{user}')
24                 userList.append(user)
25     return response.json(userList)
26
27 @app.put('/user')
28 async def updateUserInfo(req):#每次只能更新一条用户的数据
29     update_user_info = req.json
30     name = ''
31     password = ''
32     dataRate = ''
33     for k,v in update_user_info.items():
34         if(k == 'name'):
35             name = v
36         elif(k == 'password'):
37             password = v
38         elif(k == 'dataRate'):
39             dataRate = v
40     db = await aiosqlite.connect(app.config.DB_NAME)
41     sql = 'select * from user where name = \'' + name + '\'';
42     cursor = await db.execute(sql)
43     row = await cursor.fetchall()
44     if(len(row) == 0):
45         await cursor.close()
46         return response.json({"msg":"user not exist"})
47     sql = 'update user set password = \'' + password + '\', dataRate = ' + str(da
48     await db.execute(sql)
49     await db.commit()
50     await db.close()
51     return response.json({"msg":"update successful"})
52
53 @app.post('/user')
54 async def insertUserInfo(req):
55     insert_user_info = req.json
56     name = ''

```

```
57     password = ''
58     dataRate = ''
59     for k,v in insert_user_info.items():
60         if(k == 'name'):
61             name = v
62         elif(k == 'password'):
63             password = v
64         elif(k == 'dataRate'):
65             dataRate = v
66     sql = 'select * from user where name = \'' + name + '\'';'
67     db = await aiosqlite.connect(app.config.DB_NAME)
68     cursor = await db.execute(sql)
69     row = await cursor.fetchall()
70     if(len(row) != 0):
71         await cursor.close()
72         return response.json({"msg":"user already exist"})
73     sql = 'insert into user values(\'' + name + '\',' + password + '\',' + str(
74     await db.execute(sql)
75     await db.commit()
76     await db.close()
77     return response.json({"msg":"insert successful"})
78
79 @app.delete('/user')
80 async def deleteUserInfo(req):
81     delete_user_info = req.json
82     name = ''
83     for v in delete_user_info.values():
84         name = v
85         break
86     sql = 'select * from user where name = \'' + name + '\'';'
87     db = await aiosqlite.connect(app.config.DB_NAME)
88     cursor = await db.execute(sql)
89     row = await cursor.fetchall()
90     if(len(row) == 0):
91         await cursor.close()
92         return response.json({"msg":"user not exist"})
93     sql = 'delete from user where name = \'' + name + '\'';'
94     await db.execute(sql)
95     await db.commit()
96     await db.close()
97     return response.json({"msg":"delete successful"})
98
99 @app.route("/zrf")
100 async def test(request):
101     return response.json({"hello": "world"})
102
103 if __name__ == "__main__":
```

```
104 app.run(host="0.0.0.0", port=8000)
105
```



代码说明

源代码中不要出现大段的说明注释，所有文字描述在本节中以行号引用说明。

昵称

邮箱

网址(http://)

Just go go

M↓

提交

来发评论吧~