1.7 Web 学生管理程序

1.7.1 定义通讯协议

基于前面介绍过的 Flask Web 网站与 urllib 的访问网站的方法，我们设计一个学生信息管理程序，它是一个基于 Web 的学生记录管理程序。

学生的记录包括学号 No、姓名 Name、性别 Sex 与年龄 Age，服务器的作用是建立与维护一个 Sqllite 的学生数据库 students.db 中的学生记录表 students：

create table students (No varchar(16) primary key,Name varchar(16), Sex varchar(8), Age int)

服务器建立一个 Web 网站，同时提供查询学生记录、增加学生记录、删除学生记录等接口服务。服务器为了与客户端通讯，建立一个 opt 的参数如表 1-7-1 所示：

opt 值	含义
init	初始化学生表
insert	增加学生
delete	删除学生
	获取学生记录

表 1-7-1 opt 值

我们约定：

- 如果客户端向服务器发送 opt="init"，那么服务器创建 students 表，并返回是否创建成功，如果成功就返回{"msg":"OK"};
- 如果客户端向服务器发送 opt="insert"，同时发送 No,Name,Sex,Age 参数，那么服务器向数据库表插入一条学生记录，并返回是否插入成功信息，如果成功就返回{"msg":"OK"};
- 如果客户端向服务器发送 opt="delete"，同时发送 No 参数，那么服务器从数据库表中删除学号为 No 的一条学生记录，并返回是否删除成功的信息，如果成功就返回{"msg":"OK"};
- 如果客户端不向服务器发送 opt 参数值，那么服务器获取所有的学生记录返回给客户端，如果成功就返回{"msg":"OK","data":rows}，其中 rows 是学生的记录行的列表；

1.7.2 服务器程序

```
import flask
import sqlite3
import json

app=flask.Flask(__name__)

class StudentDB:
    def openDB(self):
        self.con=sqlite3.connect("students.db")
```

```
        self.cursor=self.con.cursor()

    def closeDB(self):
        self.con.commit()
        self.con.close()

    def initTable(self):
        res={}
        try:
            self.cursor.execute("create table students (No varchar(16) primary key,Name varchar(16), Sex varchar(8), Age int)")
            res["msg"]="OK"
        except Exception as err:
            res["msg"]=str(err)
        return res

    def insertRow(self,No,Name,Sex,Age):
        res={}
        try:
            self.cursor.execute("insert into students (No,Name,Sex,Age) values (?,?,?,?)",(No,Name,Sex,Age))
            res["msg"]="OK"
        except Exception as err:
            res["msg"]=str(err)
        return res

    def deleteRow(self,No):
        res={}
        try:
            self.cursor.execute("delete from students where No=?",(No,))
            res["msg"]="OK"
        except Exception as err:
            res["msg"]=str(err)
        return res

    def selectRows(self):
        res={}
        try:
            data=[]
            self.cursor.execute("select * from students order by No")
            rows=self.cursor.fetchall()
            for row in rows:
                d={}
                d["No"]=row[0]
```

```
                d["Name"]=row[1]
                d["Sex"]=row[2]
                d["Age"]=row[3]
                data.append(d)
            res["msg"]="OK"
            res["data"]=data
        except Exception as err:
            res["msg"]=str(err)
        return res

@app.route("/",methods=["GET","POST"])
def process():
    opt=flask.request.values.get("opt") if "opt" in flask.request.values else ""
    res={}
    db = StudentDB()
    db.openDB()
    if opt=="init":
        res=db.initTable()
    elif opt=="insert":
        No=flask.request.values.get("No") if "No" in flask.request.values else ""
        Name = flask.request.values.get("Name") if "Name" in flask.request.values else ""
        Sex=flask.request.values.get("Sex") if "Sex" in flask.request.values else ""
        Age = flask.request.values.get("Age") if "Age" in flask.request.values else ""
        res=db.insertRow(No,Name,Sex,Age)
    elif opt=="delete":
        No=flask.request.values.get("No") if "No" in flask.request.values else ""
        res=db.deleteRow(No)
    else:
        res=db.selectRows()
    db.closeDB()
    return json.dumps(res)

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

1.7.3 客户端程序

```
import urllib.request
import json

class Student:
    def __init__(self, No, Name, Sex, Age):
        self.No = No
        self.Name = Name
        self.Sex = Sex
```

```
        self.Age = Age

    def show(self):
        print("%-16s %-16s %-8s %-4d" % (self.No, self.Name, self.Sex, self.Age))


students = []
url = "http://127.0.0.1:5000"


def listStudents():
    global students
    print("%-16s %-16s %-8s %-4s" % ("No", "Name", "Sex", "Age"))
    for s in students:
        s.show()


def insertStudent(s):
    global students
    i = 0
    while (i < len(students) and s.No > students[i].No):
        i = i + 1
    if (i < len(students) and s.No == students[i].No):
        print(s.No + " already exists")
        return False
    students.insert(i, s)
    return True

def deleteRow():
    global students
    No = input("No=")
    if (No != ""):
        for i in range(len(students)):
            if (students[i].No == No):
                st = ""
                try:
                    st = "No=" + urllib.request.quote(No)
                    st = st.encode()
                    content = urllib.request.urlopen(url + "?opt=delete", st)
                    st = content.readline()
                    st = json.loads(st.decode())
                    st=st["msg"]
                except Exception as exp:
                    st = str(exp)
```

```
                if (st == "OK"):
                    del students[i]
                    print("删除成功")
                else:
                    print(st)
                break


def insertRow():
    No = input("No=")
    Name = input("Name=")
    while True:
        Sex = input("Sex=")
        if (Sex == "男" or Sex == "女"):
            break
        else:
            print("Sex is not valid")
    Age = input("Age=")
    if (Age == ""):
        Age = 0
    else:
        Age = int(Age)
    if No != "" and Name != "":
        s = Student(No, Name, Sex, Age)
        for x in students:
            if (x.No == No):
                print(No + " already exists")
                return
        st=""
        try:
            st = "No=" + urllib.request.quote(No) + "&Name=" + urllib.request.quote(
                Name) + "&Sex=" + urllib.request.quote(Sex) + "&Age=" + str(Age)
            st = st.encode()
            content = urllib.request.urlopen(url + "?opt=insert", st)
            st = content.read()
            st = json.loads(st.decode())
            st=st["msg"]
        except Exception as exp:
            st = str(exp)

        if (st == "OK"):
            insertStudent(s)
            print("增加成功")
```

```
            else:
                print(st)
        else:
            print("学号、姓名不能为空")


def initialize():
    st=""
    try:
        content = urllib.request.urlopen(url + "?opt=init")
        st = content.read()
        st = json.loads(st.decode())
        st=st["msg"]
    except Exception as exp:
        st=str(exp)

    if (st == "OK"):
        print("初始成功")
    else:
        print(st)
    return st

def readStudents():
    global students
    try:
        students.clear()
        content = urllib.request.urlopen(url)
        data = b""
        while True:
            buf = content.read(1024)
            if (len(buf) > 0):
                data = data + buf
            else:
                break
        data = data.decode()
        data = json.loads(data)
        if data["msg"]=="OK":
            data=data["data"]
            for d in data:
                # each d is a dictionary
                s = Student(d["No"], d["Name"], d["Sex"], d["Age"])
                students.append(s)
    except Exception as exp:
        print(exp)
```

```
try:
    readStudents()

    while True:
        print("")
        print("***学生名单***")
        print("0. 初始化学生表")
        print("1. 查看学生列表")
        print("2. 增加学生记录")
        print("3. 删除学生记录")
        print("4. 退出这个程序")
        s = input("请选择(0,1,2,3,4):")
        if (s=="0"):
            initialize()
        elif (s == "1"):
            listStudents()
        elif (s == "2"):
            insertRow()
        elif (s == "3"):
            deleteRow()
        elif (s == "4"):
            break
except Exception as exp:
    print(exp)
```

客户端结果示例：

```
***学生名单***
0. 初始化学生表
1. 查看学生列表
2. 增加学生记录
3. 删除学生记录
4. 退出这个程序
请选择(0,1,2,3,4):1
No          Name          Sex        Age
1           2             男         23
2           2             女         21
```

客户端显示有两条记录存在。