

# 用Python写一款App Inventor网络微数据库服务器

李伟 浙江省杭州市基础教育研究室

涉及学科：信息技术、教学

App Inventor是目前广受中小小学生喜爱的图形化移动应用开发软件,通过这款软件,用户可以快速地开发简易的移动应用程序。App Inventor的数据存储采用了多种方式,包括文本文件、TinyDB、TinyWebDB等,其中TinyWebDB属于网络微数据库,用户使用此数据库,可实现远程数据存储,从而实现开发网络版应用程序。

由于网络微数据库需要服务器的支持,一般的用户只能借助他人提供的远程服务器才可以使用这一功能,很不自由。在教学中,还常常因为远程服务器的连接限制而导致测试失败。因而,本文介绍一种基于Python的Flask框架开发App Inventor网络微数据库服务器的方法,让任何计算机都能“摇身”变为网络微数据库服务器,以满足个人开发和日常教学的需求。

## ● 开发网络微数据库服务器的技术分析

### 1.网络微数据库存储机制

App Inventor开发的应用

程序,是通过TinyWebDB可以完成读取数据与存储数据的功能,其相关的数据结构比较简洁,每条记录只包含了tag (键)和value (值)两项。

在利用网络微数据库进行“读取数据”时,其实质是发送一条网页form表单请求,类似于“/get?tag=tagvalue”这样的一条请求语句,传送一个tag值,服务器捕获这个请求后,查询数据库中是否存在

这个tagvalue值,假若存在,则将此值返回给APP。

而当利用网络微数据库进行“保存数据”时,其实质同样是发送一条网页form表单请求。类似于“/get? stored=storedvalue”这样的一条请求语句,传送一个stored值,此stored值实际上是一个json格式的数组字符串,即["STORED", "tag","value"],以此告知服务器程序,需要保存的数据

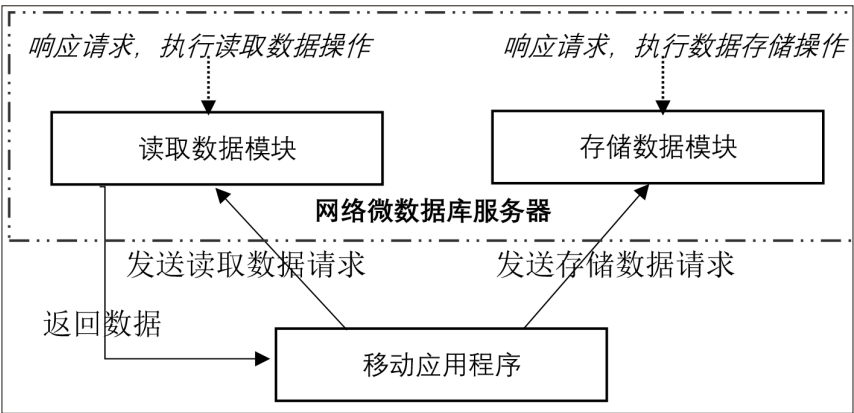


图1

网络微数据库服务器数据库结构

数据库表名称	字段名称	数据类型设计	作用
mylink	dbid	integer	主键, 自动编号
	webtag	string	记录tag值
	webvalue	string	记录value值

为:tag=“tag”,value=“value”。服务器捕获这个get请求后,将此数据作为一条记录存储进数据库。

2.网络微数据库服务器开发技术

由网络微数据库存储机制可知,服务器程序应提供两项服务,即数据读取与数据存储功能,其实

质是一个典型的Web应用服务平台。因此,只需搭建一个能提供数据读取与数据存储功能的Web应用服务平台,并且在格式上进行规范,即可替代原有的网络微数据库服务器。

3.开发环境选择

程序主体采用目前广受用户喜

爱的Python语言的Flask网页框架进行编写,数据库可以采用SQLite3模块进行构建。由于数据传输中需要进行json格式的转换,因此还需要导入json模块。

4.主要功能模块设计

网络微数据库服务器由数据库、服务器端等模块构成。

其中,数据库用于数据的存储,服务器端部分属于反馈系统的主逻辑,承载着响应APP请求,读取相应数据返回APP、捕获APP发送的数据并存储到数据库。

整体架构如上页图1所示。

● 网络微数据库服务器的

开发

1.数据库设计

数据库主要用于APP数据的存储。对于最基本的服务,可以设计一个数据表,并在数据表中设计两个字段,分别为webtag与webvalue。

本系统采用Python语言的SQLite3模块构建与操作数据库,其数据库结构设计如上页表所示。建立数据库的核心代码如图2所示。

2.网络微数据库服务器程序编写

服务器端程序采用Flask网页框架编写,主要包括以下两个主函数。

getvalue()函数:用于接收APP端读取数据请求,并返回json格式的目标数据。函数设计如图3所示。

storeavalue()函数:用于接收APP端存储数据请求,并将接收到的数据存储于数据库中。函数设计如图4所示。

● 网络微数据库服务器的部

署与应用

1.启动服务器主程序

网络微数据库服务器编写完成之后,即可在服务器计算机中以命令行的方式运行主程序:python tinywebdb.py。假设服务器地址为“192.168.31.132:8080”。

2.编写APP,并设置网络微数据库服务器地址

设计示例APP的界面如下页

```
import sqlite3
DATABASE = 'data/data.db'
def setup_db():
    db = sqlite3.connect(DATABASE)
    cur = db.cursor()
    cur.execute("CREATE TABLE IF NOT EXISTS mylink(dbid INTEGER PRIMARY KEY
    autoincrement, webtag text,webvalue text)")
    db.commit()
```

图2

```
def getvalue():
    ddtag = request.form.get('tag') #获取 APP 端数据请求中的 tag 值
    db = sqlite3.connect(DATABASE)
    cur = db.cursor()
    db.commit()
    cur.execute("SELECT * FROM mylink where webtag = '%s'" % ddtag)
    d1 = cur.fetchone()
    if d1:
        #判断数据库是否存在所请求的 tag 值
        d2 = d1[2] #如数据库中 tag 的值存在, 则返回相应的 value 值
    else:
        d2='nothing!'
    cur.close()
    db.close()
    return1 = ["VALUE", ddtag, d2]
    return2 = json.dumps(return1) #将返回值转换成 json 格式
    return return2
```

图3

```
def storeavalue():
    ddtag = request.form.get('tag') #获取 APP 端数据请求中的 tag 值
    ddvalue = request.form.get('value') #获取 APP 端数据请求中的 value 值
    db = sqlite3.connect(DATABASE)
    cur = db.cursor()
    d2value = cur.execute("select * from mylink where webtag='%s'" % ddtag)
    d2v = d2value.fetchone()
    if d2v:
        #如数据库中存在 tag 值, 则更新其 value 值, 否则新建一条记录, 存储
        tag 与 value
        cur.execute("update mylink set webvalue='%s' where webtag='%s' " %
        ( ddvalue,ddtag))
        db.commit()
    else:
        cur.execute("INSERT INTO mylink(webtag,webvalue) VALUES('%s','%s')" % (ddtag,
        ddvalue))
        db.commit()
    cur.close()
    db.close()
    return1 = ["STORED", ddtag, ddvalue ]
    return2 = json.dumps(return1)
    return return2
```

图4

**编者按:**在过去的2021年,身处科技圈的我们,总会多多少少听到“元宇宙”这个词,它是一个庞大的商业计划,但也是一个简单的哲学命题:我们是否可以复制或者超越真实的世界。回望计算机发展史,英国计算机科学家图灵做了一个重要的工作,就是提出了一个假设,相信存在一台抽象的通用计算机可以模拟任何物理过程,这个假设被多次印证之后,被称作是图灵原理。当我们将物理过程推广到生命过程或者智能行为的时候,狭义的图灵原理就被推广到广义的层面,我们可以称其为广义的图灵原理。

本年度的信息技术实验栏目的选题,仍然关注信息技术实验教学、人工智能、基于数据素养的跨学科学习等之前一直受推动和关注的领域,但是我们希望用“真实数据—模拟—仿真数据”这样一个相对统一的过程,最后通过对“真实数据”和“仿真数据”判定的不可区分性来验证图灵原理在狭义和广义层面上的正确性。这可以看作是一种计算机科学的“溯源”,也可以作为信息科技核心素养的另一种表达方式。

# 构建基于模拟的世界观:图灵原理回望

吴俊杰 北京师范大学

相对于图灵测试,很多人对图灵原理更加陌生。图灵原理来自“通用图灵机”,认为存在一种通用

的机器,可以运行任意的指令,这些指令可以计算物理问题。二战期间,在处理计算导弹弹道和破译

密码这两类不同问题的时候,都可以用同一个计算机来完成,也就是说现代计算机就是一种“通用图灵

图5所示,网络微数据库服务器地址设置如图6所示。

本示例APP只实现两个功能,即将文本框中输入的tag值与value值存入网络微数据库中;向服务器发送读取数据请求,并将相应tag值的返回值加以显示。其逻辑设计如图7所示。

### 3. 调试

在APP编写完毕后,使用AI伴侣进行测试,结果如图8所示。



图5



图6

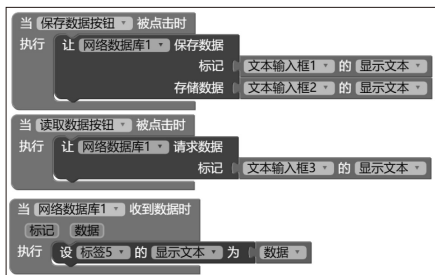


图7



图8

作者简介: 李伟 (1970.3—), 高级教师, 硕士, 主要研究中小学信息技术教育。