

# 基于 Android 和 MySQL 的 Soso 移动业务大厅

Java 作业 3-潘巧巧-1613415

## 目录

一、	整体基本实现情况.....	2
二、	实现详情.....	2
1、	远端数据库.....	2
2、	JDBC 与 DAO.....	2
	(1) DBConnection.java .....	2
	(2) 数据库接口调用实例 .....	4
三、	实现难点：多线程交互.....	5
四、	遇到的问题.....	6

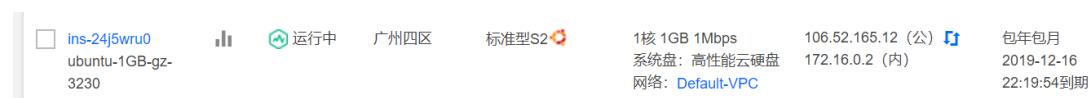
# 一、整体基本实现情况

基于本学期的 Java 作业 1 的 Soso 移动大厅进行改进，将数据同步存储至远端云上的 MySQL 中。并且为保留 Android 运行的效率，选择保留原有的本机数据结构缓存存储，即每次需要存储数据时，在本机和远端服务器上各保存一份；需要请求数据时，先在本机缓存中查看，如果没有再到远端服务器的数据库中查找。

## 二、实现详情

### 1、远端数据库

购买了一台腾讯云服务器用作数据存储。在服务器的 mysql 中新建 Databases，并将本机存储的数据结构 copy 过去。



### 2、JDBC 与 DAO

#### (1) DBConnection.java

代码中新增了一个 DBConnection.java 文件，用于提供数据库操作接口。Class 中定义了 4 个 String 和 1 个 linkMysql() 函数。4 个 String 存储了云服务器的 ip、端口、数据库表基本信息，linkMysql 函数提供了数据库操作。

linkMysql 函数有 4 个参数，第一个参数是需要执行的 sql 语句，外部操作直接提供；第二个参数是操作的枚举，指明本次操作的目的（如注册、登录）；第三个参数是本次操作

需要的一些信息。返回值是 sql 查询的结果，存放于自定义的 class ReturnData 中，包括一个序列 id 和 map 存储的 result。

```
class ReturnData{
    int id;
    Map result;
}

public class DBConnection {
    private static final String DBDRIVER = "com.mysql.jdbc.Driver";
    private static final String DBURL =
        "jdbc:mysql://59.110.140.174:3306/Soso?useUnicode=true&characterEncoding=utf-8&useSSL=false";
    private static final String DBUSER = "root";
    private static final String DBPASSWORD = "";

    public ReturnData linkMysql(String sql, OPTION op, Map<String, String> other_info) {
        Connection conn = null;
        Statement stmt = null;
        ReturnData returnData = new ReturnData();
        try {
            Class.forName(DBDRIVER);
        }
        catch (Exception e){
            e.printStackTrace();
            returnData.id = -1;
            return returnData;
        }
        try{
            conn = DriverManager.getConnection(DBURL, DBUSER, DBPASSWORD);
            stmt = conn.createStatement();
            System.out.println(sql);
            Message msg = new Message();
            switch (op) {
                case EM_REGISTER:
                    stmt.executeUpdate(sql);
                    break;
                case EM_LOGIN:
                    ResultSet rs = stmt.executeQuery(sql);
                    returnData.result = new HashMap();
                    returnData.result = other_info;
                    returnData.result.put("CorrectPassword", "null");
                    while (rs.next()) {
```

```

        returnData.result.put("CorrectPassword",rs.getString("PassWord"));
        msg.obj = returnData.result;
    }
    rs.close();
    break;
}

stmt.close();
conn.close();

msg.what = op.TurnToInt(op);
returnData.id = msg.what;
System.out.println("return db");
return returnData;
} catch (Exception e) {
    e.printStackTrace();
    returnData.id = -1;
    return returnData;
}
finally {
    if(conn!=null){
        try {
            conn.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
}
}
}
}

```

## (2) 数据库接口调用实例

以登录请求为例，在新建的线程外拼接好 sql 语句后，启动新线程，在线程里实例化 DBConnection 的对象 db,准备好参数信息后调用 linkMysql 函数,得到一个 ReturnData 结构体，对结构体中的数据进行解析。通过 Handler 进行线程间通信，回到主线程后继续进行后续操作。

```

final String dbInfo = "SELECT PassWord from MobileCardMYSQL WHERE CardNumber = \"" + GetNumber + "\"";
new Thread(new Runnable() {

```

```

public void run() {
    DBConnection db = new DBConnection();
    Map other_info = new HashMap();
    other_info.put("Number", GetNumber);
    other_info.put("UsrPassword", GetPassword);
    ReturnData returnData = db.linkMysql(dbInfo, OPTION. EM_LOGIN, other_info);
    Message msg = new Message();
    msg.what = returnData.id;
    msg.obj = returnData.result;
    uiHandler.sendMessage(msg);
}
}).start();

```

## 三、实现难点：多线程交互

在安卓开发中，需要请求数据库必须要启用多线程，而进行 UI 界面相关展示和跳转的时候必须回到主线程，否则无法展示，这就涉及到线程间交互问题，本次作业选择使用 Handler 进行交互。

```

@SuppressLint("HandlerLeak")
public Handler uiHandler = new Handler() {
    public void handleMessage(Message msg) {
        switch (msg.what) {
            case 1: // 注册
                break;
            case 2: // 登录
                UsrLogin1((Map<String, String>) msg.obj);
                break;
        }
    }
};

```

## 四、遇到的问题

(1) java.sql.SQLException: Incorrect string value: '\xE8\xAF\x9D\xE7\x97\xA8...' for column...

解决：格式不对，中文改英文

(2) at com.mysql.jdbc.StatementImpl.executeUpdateInternal

解决：由于在执行插入操作时，调用了 executeQuery 方法，出现错误；插入操作应该调用 executeUpdate 方法。  
int result = stat.executeUpdate(sql.toString());

(3) The directory Git is registered as a Git root, but no Git repositories were found there.

解决：选查看详情，删除这个不存在的项目