

文章编号: 2096-1472(2016)-10-50-04

# 基于Android和JavaEE订餐系统的设计与实现

罗国涛

(四川托普信息技术职业学院计算机系,四川 成都 611743)

**摘 要:** 针对目前手机订餐系统客户端与服务器端数据传输效率低, 以及订餐后台管理系统架构可扩展性较弱, 本文提出并实现了一种基于Android和JavaEE的手机订餐系统。订餐客户端采用Json数据格式与服务器端进行数据交互, 提高了网络数据传输效率。订餐后台管理系统采用SSH2+DWR整合框架使得展示层、业务层以及数据持久层三层分离, 降低了系统的耦合性, 提高了系统的扩展性与可维护性。实践证明, 该架构实现的订餐系统能提高餐馆或者酒店的点菜服务质量以及网上订餐效率。

**关键词:** Android; SSH2; 数据模型; Json数据格式; DWR

**中图分类号:** TP311.52 **文献标识码:** A

## Design and Implementation of the Mobile Food Ordering System Based on Android and JavaEE

LUO Guotao

(College of Computer Science, Sichuan TOPIT Vocational Institute, Chengdu 611743, China)

**Abstract:** As for the present mobile food ordering systems, the data transmission efficiency between the client and the server is quite low and the architecture expansibility of the backstage management system is poor. In view of the above, this paper proposes a mobile food ordering system based on Android and JavaEE. Json data format is adopted on the client for data interaction with the server, which improves the data transmission efficiency. The integrated framework of SSH2+DWR is utilized in the backstage management system to separate the presentation layer, business layer and data persistence layer, which reduces the coupling of the system and promotes its expansibility and maintainability. Practice has proved that the food ordering system under this architecture can effectively improve the service quality and ordering efficiency for restaurants, hotels and online food stores.

**Keywords:** Android; SSH2; data model; Json data format; DWR

## 1 引言(Introduction)

随着计算机和通信设备网路技术的不断发展, 人们对移动设备特别是Android移动终端开发出功能强大、界面美观且操作简易的应用程序提出了更高的要求<sup>[1]</sup>。JavaEE是当前比较流行的Web服务应用框架<sup>[2]</sup>, Android移动终端如何与JavaEE服务器端进行数据交互成为新的研究热点<sup>[3]</sup>。随着人们工作的繁忙、生活节奏的加快、有部分人没有充足的时间到餐馆或者酒店就餐, 他们希望通过智能手机在网上进行订餐。如何利用基于Android的智能手机终端开发移动App, 向顾客提供一种便捷、高效的餐饮服务具有重要的现实意义和广泛的应用价值。文献[4—7]虽然设计并实现了基于Android平台的订餐系统, 但服务器端采用普通的Servlet控制器与Android客户端进行数据交互, 对后期服务器的扩展具有一定的局限性, 同时服务器端与手机客户端数据交互采用xml文件格式, 数据传输效率有待提高。文献[8]虽然服务器端

采用的是WebService与手机客户端进行数据交互, 但对订餐后台管理系统的架构研究较少, 因此, 本文提出并实现了一种基于Android和JavaEE的手机订餐系统, 服务器端采用Struts2+Spring4+Hibernate4+DWR整合框架, 该整合框架可以提高后台管理系统的用户使用体验, 手机客户端与服务器端的数据交互采用JSON格式, 可以提高网络数据的传输效率。该系统是对现有订餐系统的一种补充, 对类似系统的开发具有一定的参考价值。

## 2 系统设计(Design of system)

### 2.1 系统功能总体设计

基于Android和JavaEE的订餐系统包含基于Android的手机订餐模块和基于JavaEE的订餐后台管理模块。基于Android的手机订餐模块包括顾客登录和厨师登录。顾客登录成功后可以浏览菜品、添加购物车、查看我的订单, 以及对菜品进行评价。

厨师登录成功后可以浏览订单、修改菜品制作状态。基于JavaEE的订餐后台管理模块包括系统管理、菜品类别管理、菜谱管理、订单管理、菜品留言管理、菜品评价管理和结算管理等功能。系统总体功能设计如图1所示。

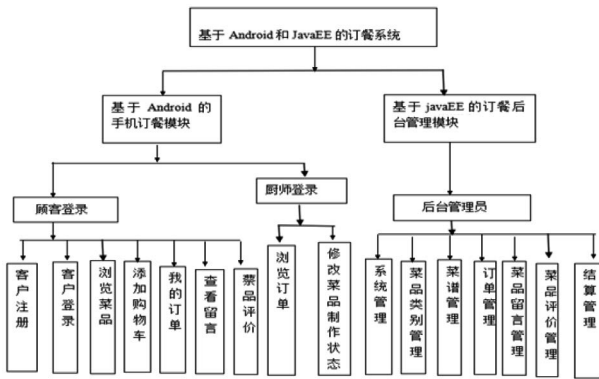


图1 系统总体功能设计

Fig.1 The overall functional design of the system

## 2.2 系统架构总体设计

系统架构总体设计采用Android手机客户端访问JavaEE服务器端的C/S架构与订餐后台管理系统通过Web浏览器访问JavaEE服务器端的B/S架构相结合的架构模式，其中，JavaEE服务器端直接访问应用数据库进行数据存取，并将数据返回给Android手机客户端和Web浏览器。系统总体架构如图2所示。

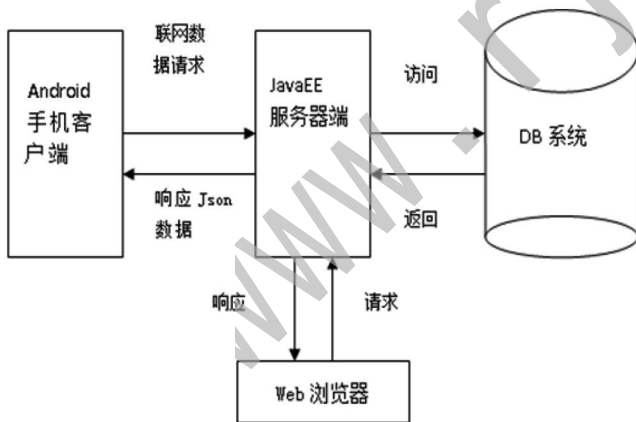


图2 系统总体设计

Fig.2 Overall system design

整个系统的数据执行流程为服务器端的订餐后台管理系统通过Web浏览器向服务器端请求存取数据操作如添加菜品类型和菜谱信息，服务器端接收请求数据后将数据存入数据库，并将存入数据库中的数据返回服务器端，服务器端再将数据返回浏览器。Android手机客户端要显示菜品信息需要联网请求服务器，服务器再根据需求访问数据库，数据库系统

处理完数据后将数据返回服务器端，服务器端将数据转换成Json格式后再返回给Android手机客户端。从整个系统数据流的执行过程来看，Android手机客户端和Web浏览器共用了JavaEE服务器端从而完成了数据的交互。

## 2.3 系统服务器架构设计

从系统总体设计的数据流执行过程来看，Android手机客户端和Web浏览器共用了同一个服务器端，因此服务器端的架构设计就显得尤为重要。在JavaEE平台下，服务器端的架构大都采用三层架构，比较典型的框架有采用MVC模式的Jsp+Servlet+JavaBean、Struts+Spring+Hibernate(SSH)和Struts2+Spring4+Hibernate4(SSH2)。其中Jsp+Servlet+JavaBean框架虽然采用MVC模式将显示层、控制层和模型层分开，但由于Servlet和JavaBean负责了大量的运算和业务处理工作，程序处理效率较为低下。SSH框架虽然也将视图层、控制器与模型的彻底分开，但由于Struts框架存在线程安全、捕获输入较难，以及视图展示都弱于Struts2框架等问题<sup>[9]</sup>，因此本系统服务器端的架构方案采用SSH2整合框架，整个系统架构设计如图3所示。

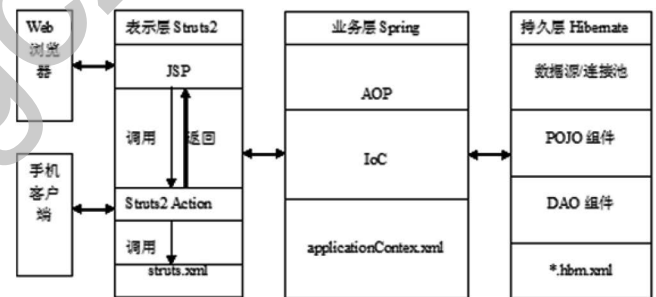


图3 系统架构设计

Fig.3 The design of system architecture

在Struts2+Spring4+Hibernate4整合框架中，Struts2主要用于Web展示和业务控制，Spring4主要用于业务实现和事务管理，而Hibernate4则用于数据持久化处理。在Struts2中Web浏览器请求服务器端的JSP页面，JSP调用Struts2中的Action完成数据的交互，而手机客户端则直接调用Struts2中的Action完成数据的交互。不论是Web端的JSP请求Struts2中的Action还是手机客户端请求Struts2中的Action，都需要通过Struts.xml配置文件实现业务处理或页面跳转等动作。Action最终将业务层的执行结果返回给Web浏览器或者手机客户端。Spring4负责处理业务逻辑及事务管理，并将Struts2层传递过来的数据依赖注入到持久层。在持久层中，Hibernate4是一个功能强大的ORM工具，它主要包括对数据源的配置、

POJO组件以及DAO组件的管理,处理DAO组件的数据,并将处理结果返回给业务层,业务层再将数据返回给表示层,最终通过浏览器或者手机客户端展示数据。

### 3 系统实现(Implementation of system)

#### 3.1 订餐系统服务器端数据封装

在基于JavaEE的订餐后台管理系统中,服务器端数据的封装既可以是JSON数据格式也可以是XML数据格式,由于JSON数据格式封装的数据在网络传输与解析中优于XML数据<sup>[10]</sup>,因此在本系统中服务器端的数据采用JSON封装。JSON即JavaScript Object Notation,它主要用于Web服务器与JavaScript或者手机客户端进行交互的一种纯文本的数据处理格式。该数据处理格式既可以处理字符串类型、数字类型、布尔类型、数组类型,也可以处理对象。在项目开发中,可以将字符串String、Object对象、JavaBean对象、List集合、数组对象、Map集合、Set集合等按照一定的算法封装成JSON格式的数据。在本系统开发中,服务器首先从数据库服务器获取数据,然后将数据存入List集合中,再将List集合中的数据封装成JSON数据格式,List集合数据封装成JSON格式数据的主要实现代码如下:

```
public static String getJson(List<?>foodList) {
    StringBuffer stringBuffer=new StringBuffer();//创建
    可改变的字符串对象stringBuffer
    //向stringBuffer对象中追加数据
    stringBuffer.append("{"");stringBuffer.
    append("\showGoodsList\":"");
    stringBuffer.append("[");
    //获取订餐列表foodList中值的大小
    int length=foodList.size();
    //将foodList中的值转换成JSON数据格式
    if(length>0)
    {
        if (foodList!=null) {
            for (Object obj:foodList) {
                stringBuffer.append(getJson(obj));
                stringBuffer.append(",");
            }
            stringBuffer.setCharAt(stringBuffer.
            length()-1,']');
            stringBuffer.append("}");
```

```
}
}
else {
    stringBuffer.append("}");
}
return stringBuffer.toString();
}
```

在上述算法中,需要首先通过菜品信息管理界面将菜品信息添加到数据库中,然后Android客户端在访问菜品信息时再从数据库中按照一定的条件查询出数据,再将查询出来的数据封装到List集合中,再将封装好的List集合数据采用本算法封装成JSON数据格式,并将该格式数据传递给Android客户端。

#### 3.2 订餐系统客户端数据解析

在Android应用系统开发中,Android客户端经常与服务端进行数据通信。由于在本系统中服务器端的数据都是封装成JSON格式的数据,因此在Android手机客户端解析JSON数据就显得尤为重要。在Android客户端解析用List类型封装的JSON格式的数据代码实现如下:

```
public List<Food>getFoodList() {
    List<Food> foods=new ArrayList<Food>();//定义
    封装菜品的集合对象
    String url=path+"showFoodList.action";//客户端请
    求服务器端的url地址
    String str=connWeb(url);//客户端请求服务器端,并
    返回具有JSON格式的字符串
    try {JSONObjectfoodjob=new JSONObject(str);//对返
    回的JSON格式的字符串封装成JSONObject对象
    JSONArray foodjay=foodjob.getJSONArray
    ("showGoodsList");//调用getJSONArray得到Json数组
    对象
    int i=0;
    int length=foodjay.length();
    while(i<length){
        JSONObject temp=(JSONObject) foodjay.get(i);//取
        出json中的每一个数组对象
        Food food=new Food();
        //取出json中对应的值并放入food对象
        food.setFprice(temp.getInt("fprice"));
```

```

food.setFoodName(temp.getString("foodName"));
food.setFcount(temp.getInt("fcount"));

foods.add(food); //将food对象放入food集合中

    i=i+1;
    }
} catch(Exception e) {
    e.printStackTrace();
}

return foods;}

```

通过上述Android客户端解析用List类型封装的JSON格式数据后,将得到的List类型数据通过SimpleAdapter适配器将数据映射到Android XML文件上,得到菜品数据列表。

### 3.3 SSH2+DWR整合框架应用

在基于JavaEE的订餐后台管理系统中,Web页面经常要与服务器端进行数据交互,而大多数Web页面希望与服务器进行数据交换时只能局部刷新页面,这样能提高用户的使用体验,而要实现Web页面局部刷新,可以采用Ajax框架,而Ajax框架有很多,如Extjs、Gwt、Dwr、JQuery等,其中DWR(Direct Web Remoting)是Getahead推出的基于java的Ajax框架,它能够将服务器端的java代码转换成JavaScript中引用的对象,从而简化客户端JavaScript的编程。为了提高Web页面中Javascript编程效率,本系统采用DWR框架与SSH2框架整合应用实现后台管理系统功能,其整合步骤如下:

(1)在web.xml文件中配置DwrServlet来加载DWR框架,部分代码如下:

```

<servlet>
    <servlet-name>dwr</servlet-name>
    <servlet-class>
org.directwebremoting.servlet.DwrServlet
</servlet-class>
    ...
</servlet>
    ...

```

(2)在WEB-INF工程下面创建dwr.xml文件,在该文件中通过creator属性定义服务器端的java类的产生方式。如果creator="new",则直接调用普通的类,如果creator="spring",则通过Spring文件中的bean组件找到需要调用的java类,并将该java类生成JavaScript代码供jsp调

用。在本系统中dwr采用Spring文件的方式将java类生成JavaScript的部分代码如下:

```

<dwr>
    <allow>
        <create creator="spring"javascript="DWRUser">
            <param name="beanName" value="duser"/> //定义
调用bean的id值为duser
            <include method="getUser"/> //定义调用java类中的
getUser方法

```

```

        </create>
    </allow>
</dwr>

```

在该代码中,spring文件中bean组件所对应的代码为:

```

<bean id="duser" class="com.food.serviceImpl.
UserServiceImpl"></bean>

```

而UserServiceImpl所对应的java类代码如下:

```

public class UserService Implements IUserService{
    public String getUser(String name)//该方法
也就是javascript调用的方法
    {...}
}

```

在jsp中要用dwr框架来调用服务器端产生的JavaScript代码,需要在jsp中导入DWR核心库文件util.js、engine.js,以及动态生成的js库文件DWRUser.js。Jsp中通过DWR调用java类中的方法的部分代码如下:

```

<script type="text/javascript">
    function checkUser()
    {DWRUser.getUser("lgt",callBack); //调用
UserServiceImpl中的getUser方法
    }
    function callBack(data)
    {...}
</script>

```

## 4 结论(Conclusion)

本文提出了基于Android和JavaEE订餐系统的设计与实现,整个系统由Android手机客户端、JavaEE服务器端和数据库三部分组成。JavaEE服务器端采用SSH2整合框架将服务器端的架构分为展示层、业务逻辑层以及数据持久层,这样不

(下转第43页)