《移动终端开发技术》

电子拍卖系统

课程名称：移动终端开发技术

所属系部：电子信息工程学院

制定人：曹洪新

合作人：

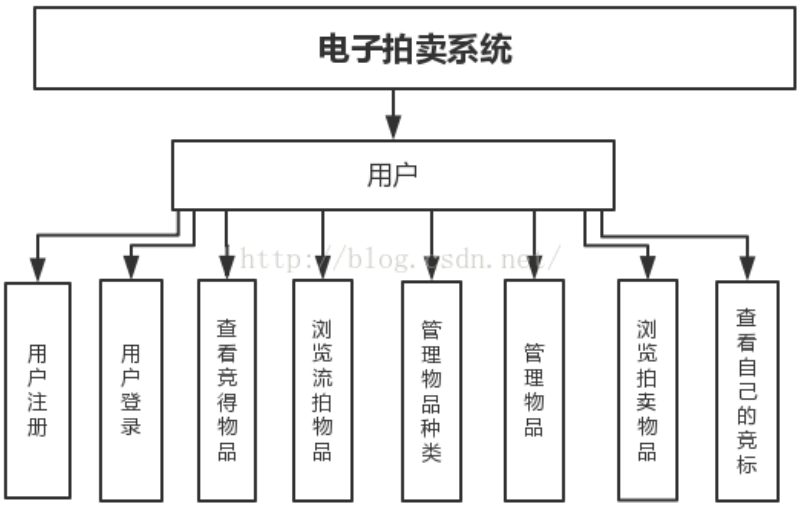
制定时间：2018 年 8 月

日照职业技术学院

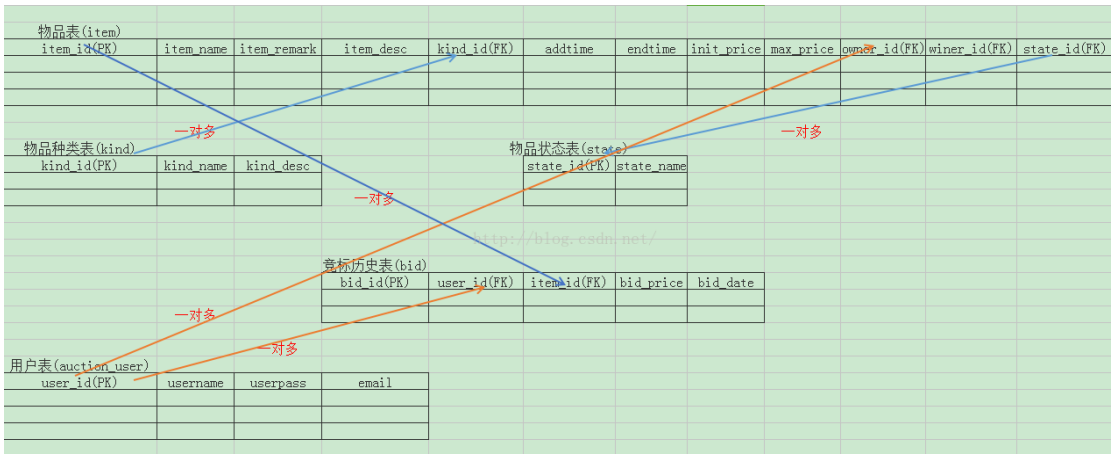
目 录

第一章　电子拍卖系统设计

1.1 功能流程图



1.2 数据库设计



说明：

物品表(item)、物品种类表(kind)、物品状态表(state)、竞价历史表(bid)、用户表(auction_user)

表之间的关系：

物品种类与物品：一对多，物品与物品状态：一对多，物品与竞价历史：一对多

用户与竞价历史：一对多，用户与物品：一对多

1.3 SQL 语句创建

```
drop database if exists auction;
create database auction;
use auction;
#用户表
create table auction_user(
  user_id int(11) auto_increment,
  username varchar(50) not null,
  userpass varchar(50) not null,
email varchar(100) not null,
  primary key(user_id),
  unique(username)
);
INSERT INTO auction_user (username,userpass,email) VALUES ('tomcat','
tomcat','spring_test@163.com');
INSERT INTO auction_user (username,userpass,email) VALUES ('mysql','m
ysql','spring_test@163.com');
#物品种类表
create table kind(
  kind_id int(11) auto_increment,
  kind_name varchar(50) not null,
  kind_desc varchar(255) not null,
  primary key(kind_id)
);

INSERT INTO kind (kind_name,kind_desc) VALUES ('电脑硬件','这里并不是
很主流的产品，但价格绝对令你心动');
INSERT INTO kind (kind_name,kind_desc) VALUES ('房产','提供非常稀缺的
房源');
```

```
#物品状态表
create table state(
  state_id int(11) auto_increment,
  state_name varchar(10),
  primary key(state_id)
);
INSERT INTO state (state_name) VALUES ('拍卖中');
INSERT INTO state (state_name) VALUES ('拍卖成功');
INSERT INTO state (state_name) VALUES ('流拍');
#物品表
create table item(
  item_id int(11) auto_increment,
  item_name varchar(255) not null,
  item_remark varchar(255),
item_desc varchar(255),
  kind_id int(11) not null,
  addtime date not null,
  endtime date not null,
  init_price double not null,
  max_price double not null,
  owner_id int(11) not null,
  winer_id int(11),
  state_id int(11) not null,
  primary key(item_id),
  FOREIGN KEY(kind_id) REFERENCES kind(kind_id),
  FOREIGN KEY(owner_id) REFERENCES auction_user(user_id),
  FOREIGN KEY(winer_id) REFERENCES auction_user(user_id),
  FOREIGN KEY(state_id) REFERENCES state(state_id)
);
```

```
# 拍卖中的物品
INSERT INTO item ( item_name , item_remark , item_desc, kind_id, addtime , endtime, init_price, max_price, owner_id, winer_id, state_id)
    VALUES ( '主板', '老式主板', '老主板，还可以用', 1, ADDDATE(CURDATE(), -5), ADDDATE(CURDATE(), 30) , 230, 250, 1, null, 1);
# 流派的物品
INSERT INTO item ( item_name , item_remark , item_desc, kind_id, addtime , endtime, init_price, max_price, owner_id, winer_id, state_id)
VALUES ( '显卡', '老式显卡', '老显卡,还可以用', 1, ADDDATE(CURDATE(), -9), ADDDATE(CURDATE(), -2), 210, 210, 2, null, 3);
# 被竞得的物品
INSERT INTO item ( item_name , item_remark , item_desc, kind_id, addtime , endtime, init_price, max_price, owner_id, winer_id, state_id)
    VALUES ( '老房子', '老式房子', '40 年的老房子', 2, ADDDATE(CURDATE(), -9), ADDDATE(CURDATE(), -5), 21000, 25000, 2, 1, 2);
#竞标历史表
create table bid(
  bid_id int(11) auto_increment,
  user_id int(11) not null,
  item_id int(11) not null,
  bid_price double not null,
  bid_date date not null,
  primary key(bid_id),
  unique(item_id , bid_price),
FOREIGN KEY(user_id) REFERENCES auction_user(user_id),
  FOREIGN KEY(item_id) REFERENCES item(item_id)
```

```
);
INSERT INTO bid ( user_id , item_id , bid_price, bid_date)
    VALUES ( 2, 1, 250, ADDDATE(CURDATE(), -2));
INSERT INTO bid ( user_id , item_id , bid_price, bid_date)
    VALUES ( 1, 3, 25000, ADDDATE(CURDATE(), -6));
```

1.4 所需实现的功能

对于物品的管理，本系统可以查询拍卖物品，添加拍卖物品，增加物品种类，竞价处理以及发送邮件通知用户所参与的竞价。

(1) 注册用户可以添加用户物品，添加物品种类。添加之前必须登录系统。

(2) 注册用户可以浏览当前拍卖中的物品，以及流拍的物品。

(3) 注册用户可以参与竞价，参与的竞价系统将通过邮件通知用户。

第二章 设计 web 服务器

2.1 PO 对象的创建

PO(persistant object) 持久对象

通常对应数据模型(数据库),本身还有部分业务逻辑的处理。可以看成是与数据库中的表相映射的 java 对象。最简单的 PO 就是对应数据库中某个表中的一条记录，多个记录可以用 PO 的集合。PO 中应该不包含任何对数据库的操作。

User.java

```
public class User implements Serializable{
    private Integer user_id;
    private String username;
    private String userpass;
    private String email;
}
```

Item.java

```
public class Item implements Serializable {
    // 标识属性
    private Integer item_id;
    // 物品名称
    private String item_name;
    // 物品 Remark
    private String item_remark;
  // 物品描述
    private String item_desc;
    // 物品添加时间
    private Date addtime;
    // 物品结束拍卖时间
    private Date endtime;
    // 物品的起拍价
```

```
    private double init_price;
    // 物品的最高价
    private double max_price;
    // 该物品的所有者
    private User owner;
    // 该物品所属的种类
    private Kind kind;
    // 该物品的赢取者
    private User winer;
    // 该物品所处的状态
    private State itemState;
    // 该物品对应的全部竞价记录
    private Set<Bid> bids = new HashSet<Bid>();
}
```

Kind.java

```
public class Kind implements Serializable{
    // 标识属性
    private Integer kind_id;
    // 种类名
    private String kind_name;
    // 种类描述
    private String kind_desc;
    // 该种类下的所有物品
    private Set<Item> items = new HashSet<Item>();
}
```

State.java

```
public class State implements Serializable{
    // 标识属性
    private Integer state_id;
```

```
    // 物品的状态名
    private String state_name;
    // 该状态下的所有物品
    private Set<Item> items = new HashSet<Item>();
}
```

Bid.java

```
public class Bid implements Serializable{
    // 标识属性
    private Integer bid_id;
    // 竞价的价格
    private double bid_price;
    // 竞价的日期
    private Date bid_date;
    // 本次竞价所竞拍的物品
    private Item item_id;
    // 参与竞价的用户
    private User user_id;}
```

2.2 VO 对象的创建

VO(value object) 值对象

通常用于业务层之间的数据传递，和 PO 一样也是仅仅包含数据而已。但应是抽象出的业务对象,可以和表对应,也可以不,这根据业务的需要.个人觉得同 DTO(数据传输对象),在 web 上传递。

ItemBean.java

```
package com.xbmu.business;
import java.util.Date;
public class ItemBean{
    private Integer id;
    private String name;
    private String desc;
```

```
    private String remark;
    private String kind;
    private String owner;
    private String winer;
    private String state;
    private double initPrice;
    private double maxPrice;
    private Date addTime;
    private Date endTime;

    // 无参数的构造器
    public ItemBean()
    {
    }
    // 初始化全部属性的构造器
    public ItemBean(Integer id , String name , String desc , String r
emark,String kind , String owner , String winer , String state ,doubl
e initPrice , double maxPrice , Date addTime , Date endTime)
    {
        this.id = id;
        this.name = name;
        this.desc = desc;
        this.remark = remark;
        this.kind = kind;
        this.owner = owner;
        this.winer = winer;
        this.state = state;
        this.initPrice = initPrice;
        this.maxPrice = maxPrice;
```

```
        this.addTime = addTime;
        this.endTime = endTime;
    }
//...省略 set 和 get 方法
}
```

BidBean.java

```
package com.xbmu.business;
import java.io.Serializable;
import java.util.Date;
public class BidBean implements Serializable
{
    private int id;
    private String user;
    private String item;
    private double price;
    private Date bidDate;
    // 无参数的构造器
    public BidBean()
    {
    }
    // 初始化全部属性的构造器
    public BidBean(int id , String user , String item ,
        double price , Date bidDate)
    {
        this.id = id;
        this.user = user;
        this.item = item;
        this.price = price;
        this.bidDate = bidDate;
```

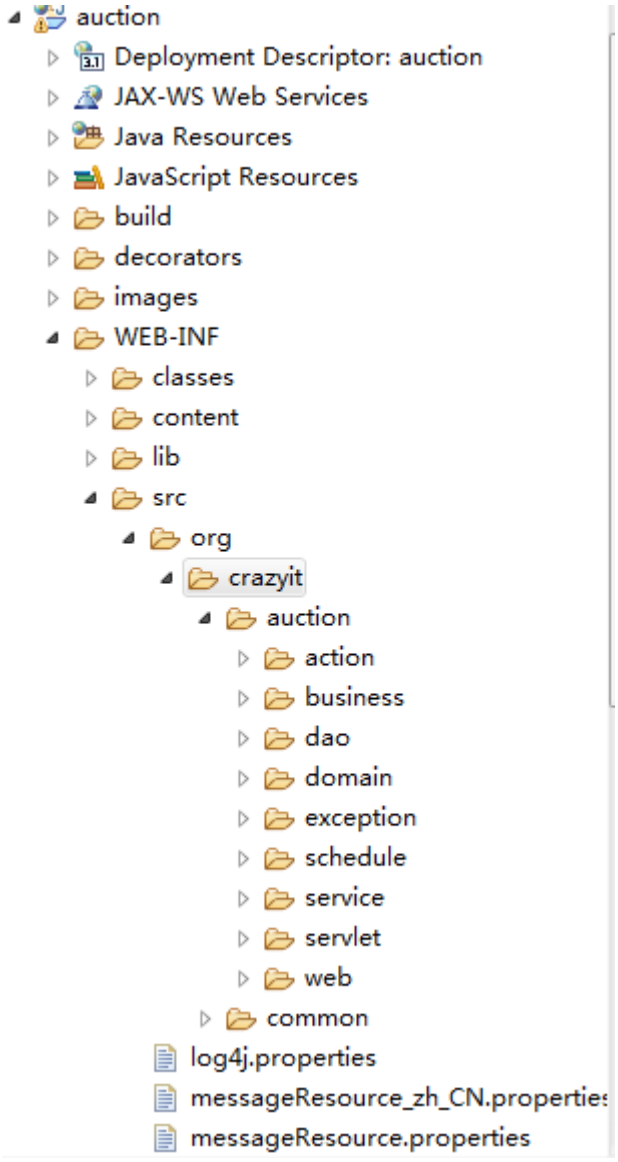
```
    }
//...
}
```

KindBean.java

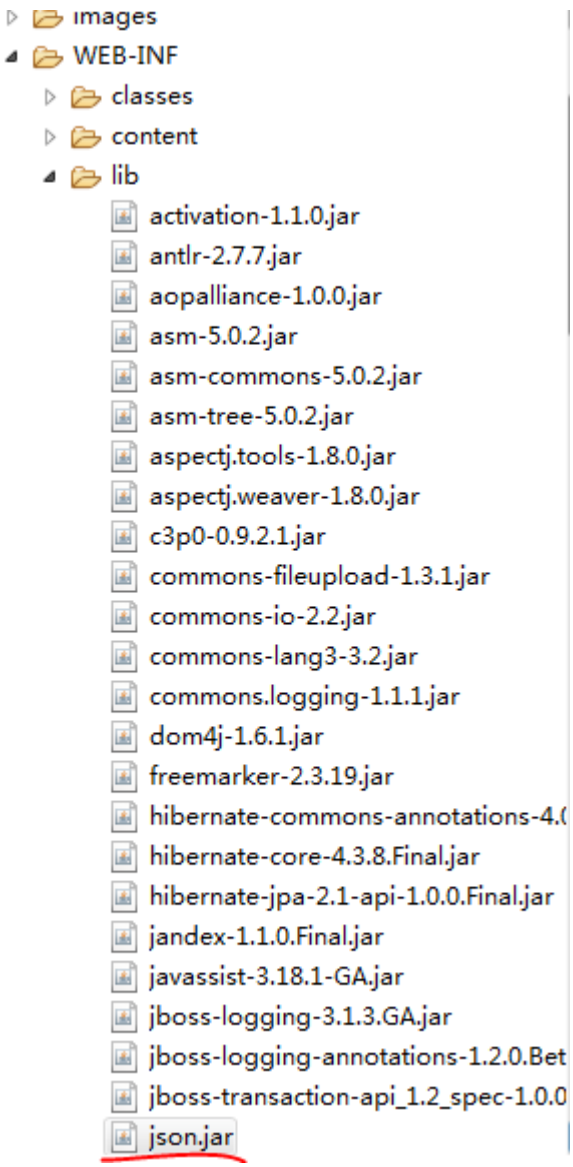
```
package com.xbmu.business;
public class KindBean{
    private int id;
    private String kindName;
    private String kindDesc;
    // 无参数的构造器
    public KindBean()
    {
    }
    // 初始化全部属性的构造器
    public KindBean(int id , String kindName , String kindDesc)
    {
        this.id = id;
        this.kindName = kindName;
        this.kindDesc = kindDesc;
    }
```

第三章 搭建简单的 web 服务器

首先简单看一下 web 工程目录及其包的构建：



导入 jar 包



- 业务层接口：AcutionManager.java

```
package org.crazyit.auction.service;
import java.util.List;
import org.crazyit.auction.business.*;
import org.crazyit.auction.dao.*;
import org.crazyit.auction.domain.*;
import org.crazyit.auction.exception.AuctionException;
/**
 * Description:
 * <br/>网站: <a href="http://www.crazyit.org">疯狂 Java 联盟</a>
 * <br/>Copyright (C), 2001-2016, Yeeku.H.Lee
```

```
 * <br/>This program is protected by copyright laws.
 * <br/>Program Name:
 * <br/>Date:
 * @author Yeeku.H.Lee kongyeeku@163.com
 * @version 1.0
 */
public interface AuctionManager
{
    /**
     * 根据赢取者查询物品
     * @param winerId 赢取者的 ID
     * @return 赢取者获得的全部物品
     */
    List<ItemBean> getItemByWiner(Integer winerId)
        throws AuctionException;
    /**
     * 查询流拍的全部物品
     * @return 全部流拍物品
     */
    List<ItemBean> getFailItems()throws AuctionException;

    /**
     * 根据用户名，密码验证登录是否成功
     * @param username 登录的用户名
     * @param pass 登录的密码
     * @return 登录成功返回用户 ID，否则返回-1
     */
    int validLogin(String username , String pass)
        throws AuctionException;
```

```
/**
 * 查询用户的全部出价
 * @param userId 竞价用户的 ID
 * @return 用户的全部出价
 */
List<BidBean> getBidByUser(Integer userId)
    throws AuctionException;

/**
 * 根据用户查找目前仍在拍卖中的全部物品
 * @param userId 所属者的 ID
 * @return 属于当前用户的、处于拍卖中的全部物品。
 */
List<ItemBean> getItemsByOwner(Integer userId)
    throws AuctionException;

/**
 * 查询全部种类
 * @return 系统中全部全部种类
 */
List<KindBean> getAllKind() throws AuctionException;

/**
 * 添加物品
 * @param item 新增的物品
 * @param avail 有效天数
 * @param kindId 物品种类 ID
 * @param userId 添加者的 ID
```

```
     * @return 新增物品的主键
     */
    int addItem(Item item, int avail , int kindId , Integer userId)
        throws AuctionException;

    /**
     * 添加种类
     * @param kind 新增的种类
     * @return 新增种类的主键
     */
    int addKind(Kind kind) throws AuctionException;

    /**
     * 根据产品分类，获取处于拍卖中的全部物品
     * @param kindId 种类 id;
     * @return 该类的全部产品
     */
    List<ItemBean> getItemsByKind(int kindId) throws AuctionExceptio
n;

    /**
     * 根据种类 id 获取种类名
     * @param kindId 种类 id;
     * @return 该种类的名称
     */
    String getKind(int kindId) throws AuctionException;

    /**
     * 根据物品 id，获取物品
```

```
     * @param itemId 物品 id;
     * @return 指定 id 对应的物品
     */
    ItemBean getItem(int itemId) throws AuctionException;

    /**
     * 增加新的竞价，并对竞价用户发邮件通知
     * @param itemId 物品 id;
     * @param bid 竞价
     * @param userId 竞价用户的 ID
     * @return 返回新增竞价记录的 ID
     */
    int addBid(int itemId , Bid bid ,Integer userId)
        throws AuctionException;

    /**
     * 根据时间来修改物品的赢取者
     */
    void updateWiner()throws AuctionException;
}
```

- 完成登录功能模块：

AcutionManagerImpl.java

```
package org.crazyit.auction.service.impl;
import org.apache.log4j.Logger;
import java.util.*;
import org.crazyit.auction.business.*;
import org.crazyit.auction.dao.*;
import org.crazyit.auction.domain.*;
import org.crazyit.auction.exception.AuctionException;
```

```
import org.crazyit.auction.service.AuctionManager;
import org.springframework.mail.MailException;
import org.springframework.mail.MailSender;
import org.springframework.mail.SimpleMailMessage;

/**
 * Description:
 * <br/>网站: <a href="http://www.crazyit.org">疯狂 Java 联盟</a>
 * <br/>Copyright (C), 2001-2016, Yeeku.H.Lee
 * <br/>This program is protected by copyright laws.
 * <br/>Program Name:
 * <br/>Date:
 * @author Yeeku.H.Lee kongyeeku@163.com
 * @version 1.0
 */
public class AuctionManagerImpl implements AuctionManager
{   //----------------------用户----------------------------
    private AuctionUserDao userDao = new AuctionUserDaoImpl();
    /**
     * 根据用户名, 密码验证登录是否成功
     * @param username 登录的用户名
     * @param pass 登录的密码
     * @return 登录成功返回用户 ID, 否则返回-1
     */
    public int validLogin(String username , String pass) throws Aucti
onException
    {
        try
        {
```

```
            AuctionUser u = userDao.findUserByNameAndPass(username , p
ass);
            if (u != null)
            {
                return u.getId();
            }
            return -1;
        }
        catch (Exception e)
        {
            log.debug(e.getMessage());
            throw new AuctionException("处理用户登录出现异常,请重试");
        }
    }
    //...
}
```

- AuctionUserDao.java

```
package org.crazyit.auction.dao;
import java.util.List;
import org.crazyit.common.dao.*;
import org.crazyit.auction.domain.*;
import org.crazyit.auction.business.*;
/**
 * Description:
 * <br/>网站: <a href="http://www.crazyit.org">疯狂 Java 联盟</a>
 * <br/>Copyright (C), 2001-2016, Yeeku.H.Lee
 * <br/>This program is protected by copyright laws.
 * <br/>Program Name:
 * <br/>Date:
```

```
 * @author Yeeku.H.Lee kongyeeku@163.com
 * @version 1.0
 */
public interface AuctionUserDao extends BaseDao<AuctionUser>
{
    /**
     * 根据用户名，密码查找用户
     * @param username 查询所需的用户名
     * @param pass 查询所需的密码
     * @return 指定用户名、密码对应的用户
     */
    AuctionUser findUserByNameAndPass(String username , String pass);
}
```

- AuctionUserDaoImpl.java

```
package org.crazyit.auction.dao.impl;
import java.util.*;
import org.crazyit.common.dao.impl.*;
import org.crazyit.auction.domain.*;
import org.crazyit.auction.business.*;
import org.crazyit.auction.dao.*;
/**
 * Description:
 * <br/>网站: <a href="http://www.crazyit.org">疯狂 Java 联盟</a>
 * <br/>Copyright (C), 2001-2016, Yeeku.H.Lee
 * <br/>This program is protected by copyright laws.
 * <br/>Program Name:
 * <br/>Date:
 * @author Yeeku.H.Lee kongyeeku@163.com
 * @version 1.0
```

```
 */
public class AuctionUserDaoHibernate
    extends BaseDaoHibernate4<AuctionUser> implements AuctionUserDao
{

    /**
     * 根据用户名，密码查找用户
     * @param username 查询所需的用户名
     * @param pass 查询所需的密码
     * @return 指定用户名、密码对应的用户
     */
    public AuctionUser findUserByNameAndPass(String username , String
 pass)
    {
        // 执行 HQL 查询
        List<AuctionUser> ul = (List<AuctionUser>)find(
           "from AuctionUser au where au.username=?0 and au.userpass=?
1" ,
            username , pass);
        // 返回查询得到的第一个 AuctionUser 对象
        if (ul.size() == 1)
        {
            return (AuctionUser)ul.get(0);
        }
        return null;
    }
}
```

- LoginServlet.java

```
package org.crazyit.auction.servlet;
```

```
import javax.servlet.*;
import javax.servlet.http.*;
import javax.servlet.annotation.*;

import org.crazyit.auction.servlet.base.BaseServlet;
import org.crazyit.auction.service.AuctionManager;
import java.io.*;
import org.json.*;

/**
 * Description:
 * <br/>网站: <a href="http://www.crazyit.org">疯狂 Java 联盟</a>
 * <br/>Copyright (C), 2011-2016, Yeeku.H.Lee
 * <br/>This program is protected by copyright laws.
 * <br/>Program Name:
 * <br/>Date:
 * @author  Yeeku.H.Lee kongyeeku@163.com
 * @version  1.0
 */
@WebServlet(urlPatterns="/android/login.jsp")
public class LoginServlet extends BaseServlet
{
    public void service(HttpServletRequest request ,
        HttpServletResponse response)
        throws IOException , ServletException
    {
        String user = request.getParameter("user");
        String pass = request.getParameter("pass");
```

```
        // 获取系统的业务逻辑组件
        AuctionManager auctionManager = (AuctionManager)getCtx().getBe
an("mgr");
        // 验证用户登录
        int userId = auctionManager.validLogin(user , pass);
        response.setContentType("text/html; charset=GBK");
        // 登录成功
        if (userId > 0)
        {
            request.getSession(true).setAttribute("userId" , userId);
        }
        try
        {
            // 把验证的 userId 封装成 JSONObject
            JSONObject jsonObj = new JSONObject()
                .put("userId" , userId);
            // 输出响应
            response.getWriter().println(jsonObj.toString());
        }
        catch (JSONException ex)
        {
            ex.printStackTrace();
        }
    }
}
```

- login.jsp

```
<%--
网站: <a href="http://www.crazyit.org">疯狂 Java 联盟</a>
author  yeeku.H.lee kongyeeku@163.com
```

```
version  1.0
Copyright (C), 2001-2012, yeeku.H.Lee
This program is protected by copyright laws.
Program Name:
Date:
--%>

<%@ page contentType="text/html; charset=GBK" language="java" errorPa
ge="" %>
<%@taglib prefix="s" uri="/struts-tags"%>
<!DOCTYPE html>
<html>
<head>
    <meta name="author" content="Yeeku.H.Lee(CrazyIt.org)" />
    <meta http-equiv="Content-Type" content="text/html; charset=GBK"
/>
    <title>商业拍卖 Java EE 程序框架</title>
</head>
<body>
<table width="780" align="center" cellspacing="0"
    background="images/bodybg.jpg">
<tr>
<td >
<h3>请输入用户名和密码登录系统</h3>
<div align="center">
<!-- 输出 Action 的错误提示 -->
<s:actionerror cssClass="error"/>
<s:form action="proLogin">
    <s:textfield name="username" label="用户名"/>
```

```
    <s:textfield name="password" label="密码"/>
    <s:textfield name="vercode" label="验证码"/>
    <s:submit value="登录"/>
</s:form>
验证码: <img name="d" src="auth.jpg">
</div>
</td>
</tr>
</table>
</body>
</html>
```

请求地址:

服务端:

登录界面: http://localhost:8080/auction/index.jsp

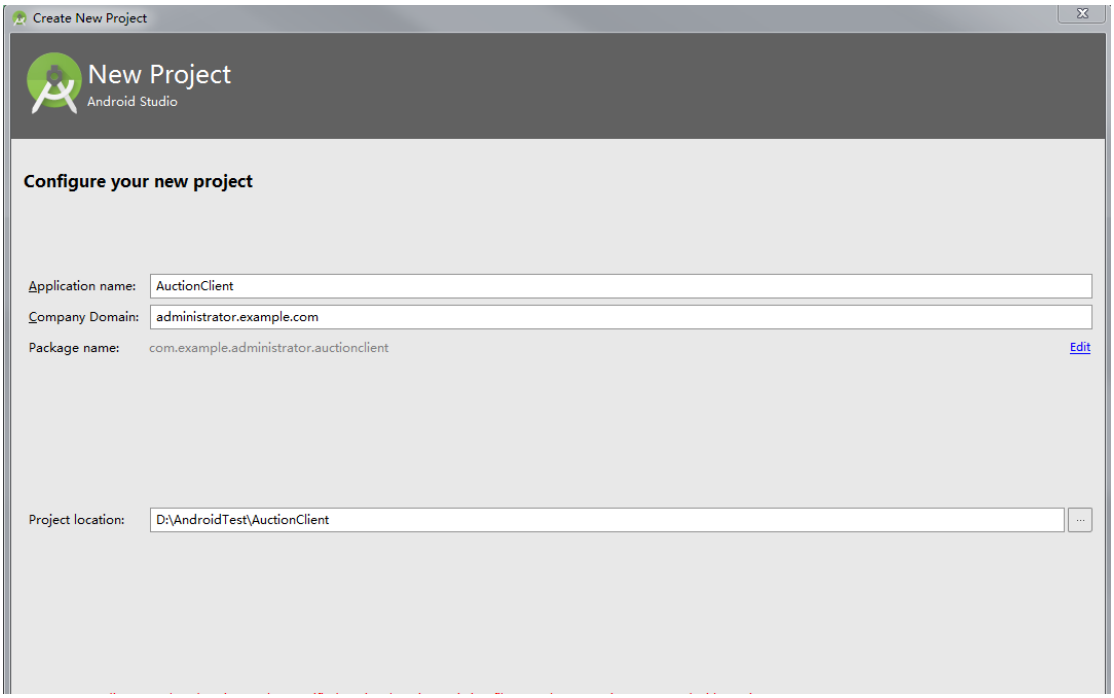
客户端:

登录访问地址:

http://192.168.1.9:8080/acutin/android/index.jsp

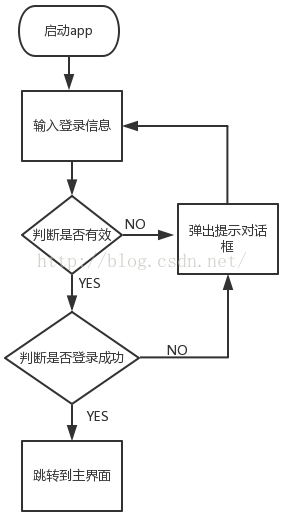
第四章 客户端开发

搭建开发环境，创建项目



4.1 登录模块

登录流程图：



FutureTask 类介绍：

FutureTask 是一种可以取消的异步的计算任务。它的计算是通过 Callable 实现的，它等价于可以携带结果的 Runnable，并且有三个状态：等待、运行和完成。

完成包括所有计算以任意的方式结束，包括正常结束、取消和异常。

Future 有个 get 方法而获取结果只有在计算完成时获取，否则会一直阻塞直到任务转入完成状态，然后会返回结果或者抛出异常。

FutureTask 有下面几个重要的方法：

1.get()

阻塞一直等待执行完成拿到结果

2.get(int timeout, TimeUnit timeUnit)

阻塞一直等待执行完成拿到结果，如果在超时时间内，没有拿到抛出异常

3.isCancelled()

是否被取消

4.isDone()

是否已经完成

5.cancel(boolean mayInterruptIfRunning)

试图取消正在执行的任务

- 访问网络的工具类 HttpUtil.java

```
package com.example.administrator.auctionclient.util;
import org.apache.http.HttpResponse;
import org.apache.http.NameValuePair;
import org.apache.http.client.HttpClient;
import org.apache.http.client.entity.UrlEncodedFormEntity;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.impl.client.DefaultHttpClient;
import org.apache.http.message.BasicNameValuePair;
import org.apache.http.util.EntityUtils;
import java.util.ArrayList;
```

```
import java.util.List;
import java.util.Map;
import java.util.concurrent.Callable;
import java.util.concurrent.FutureTask;

/**
 * Created by Administrator on 2018-8-15 0015.
 */
public class HttpUtil {
    //创建 HttpClient 对象
    public static HttpClient httpClient=new DefaultHttpClient();
    public static final String BASE_URL="http://192.168.1.9:8080/auct
ion/android/";

    public static String getRequest(final String url)throws Exception
{
        FutureTask<String> task=new FutureTask<String>(
                new Callable<String>() {
                    @Override
                    public String call() throws Exception {
                        //创建 HttpGet 对象
                        HttpGet get=new HttpGet(url);
                        //发送 get 请求
                        HttpResponse httpResponse=httpClient.execute
(get);
                        //如果服务器成功的返回响应
                        if (httpResponse.getStatusLine().getStatusCod
e()==200)
                        {
```

```
                            //获取服务器响应的字符串
                            String result= EntityUtils.toString(httpR
esponse.getEntity());
                            return result;
                        }
                        return null;
                    }
                }
        );
        new Thread(task).start();
        return task.get();
    }
    public static String postRequest(final String url, final Map<Stri
ng,String > rawparam) throws Exception
    {
        FutureTask<String> task=new FutureTask<String>(
                new Callable<String>() {
                    @Override
                    public String call() throws Exception {
                        //创建 httppost 对象
                        HttpPost post=new HttpPost(url);
                        //如果传递参数个数比较多,可以对传递的参数进行
封装
                        List<NameValuePair> params=new ArrayList<Name
ValuePair>();
                        for (String key:rawparam.keySet())
                        {
                            //封装请求参数
```

```
                params.add(new BasicNameValuePair(key,raw
param.get(key)));
                }
                //设置请求参数
                post.setEntity(new UrlEncodedFormEntity(param
s,"gbk"));
                //发送 post 请求
                HttpResponse httpResponse=httpClient.execute
(post);
                //如果服务器成功返回响应
                if (httpResponse.getStatusLine().getStatusCod
e()==200)
                {
                    String result=EntityUtils.toString(httpRe
sponse.getEntity());
                    return result;
                }
                return null;
            }
        }
    );
    new Thread(task).start();
    return task.get();
  }
}
```

- Login.java

```
package com.example.administrator.auctionclient;

import android.app.Activity;
```

```
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

import com.example.administrator.auctionclient.util.DialogUtil;
import com.example.administrator.auctionclient.util.HttpUtil;

import org.json.JSONObject;

import java.util.HashMap;
import java.util.Map;
/**
 * Created by Administrator on 2018-8-15 0015.
 */
public class Login extends Activity {
    // 定义界面中两个文本框
    EditText etName, etPass;
    // 定义界面中两个按钮
    Button bnLogin, bnCancel;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.login);
        // 获取界面中两个编辑框
        etName = (EditText) findViewById(R.id.userEditText);
        etPass = (EditText) findViewById(R.id.pwdEditText);
        // 获取界面中的两个按钮
```

```
        bnLogin = (Button) findViewById(R.id.bnLogin);
        bnCancel = (Button) findViewById(R.id.bnCancel);

        // 为 bnCancal 按钮的单击事件绑定事件监听器
        bnCancel.setOnClickListener(new HomeListener(this));
        bnLogin.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                //Toast.makeText(Login.this, "你点击了登录按钮", Toas
t.LENGTH_SHORT).show();
                //首先对用户名和密码进行校验，然后连接服务器进行认证
                if (Validate())
                {
                    //校验成功，连接服务器认证
                    if (loginPro())
                    {

                        Intent intent=new Intent(Login.this,MainActiv
ity.class);
                        startActivity(intent);
                        finish();

                    }
                    else
                    {
                        DialogUtil.showDialog(Login.this,"用户名或者
密码错误，请重新输入",false);
                    }
                }
```

```
            }
        });
    }
    //对用户名和密码进行校验
    private boolean Validate()
    {
        String username=etName.getText().toString().trim();
        if (username.equals(""))
        {

            DialogUtil.showDialog(Login.this,"用户名是必须填写的内容!
",false);
            return  false;
        }
        String pwd=etPass.getText().toString().trim();
        if (pwd.equals(""))
        {
            DialogUtil.showDialog(Login.this,"密码是必须要填写的内容!
",false);
            return false;
        }
        return true;
    }
    private boolean loginPro()
    {
        String username=etName.getText().toString();
        String pwd=etPass.getText().toString();
        JSONObject jsonobj;
        Map<String,String> map=new HashMap<>();
```

```
        map.put("user",username);
        map.put("pass",pwd);
        //定义发送请求的 URL
        String url= HttpUtil.BASE_URL+"login.jsp";
        try{
        jsonobj=new JSONObject(HttpUtil.postRequest(url,map));
            if (jsonobj.getInt("userId")>0)
                return  true;
    }catch (Exception e)
        {
            DialogUtil.showDialog(Login.this,"服务器异常,请稍后再试",
false);
            e.printStackTrace();
        }
        return false;
    }

}
```

登录界面使用了表格布局，我们这里简单介绍下表格布局的常用属性：

android:shrinkColumns：设置允许被收缩的列的列序号。多个序列号之间用逗号隔开。

android:stretchColumns：设置允许被拉伸的列的列序号。多个序列号之间用逗号隔开。

android:collapseColumns：设置需要被隐藏的列的列序号。多个序列号之间用逗号隔开。

- login.xml

```
<?xml version="1.0" encoding="utf-8"?>
<TableLayout xmlns:android="http://schemas.android.com/apk/res/androi
d"
```

```
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_gravity="center_horizontal"
            android:stretchColumns="1">
    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:scaleType="fitCenter"
        android:src="@drawable/logo"
        android:contentDescription="Hello World, Login!"/>
    <TextView
        android:text="欢迎使用电子拍卖系统"
        android:id="@+id/TextView"
        android:textSize="20dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:padding="30dp"/>
    <!-- 输入用户名的行 -->
    <TableRow>
        <TextView
            android:text="用户账户： "
            android:textSize="20dp"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"/>
        <EditText
            android:id="@+id/userEditText"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
```

```
        android:inputType="text"/>
</TableRow>
<!-- 输入密码的行 -->
<TableRow>
    <TextView
        android:text="用户口令: "
        android:textSize="20dp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>
    <EditText
        android:text=""
        android:id="@+id/pwdEditText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="textPassword"/>
</TableRow>
<!-- 定义登录、取消按钮的行 -->
<LinearLayout android:orientation="horizontal"
              android:layout_width="wrap_content"
              android:layout_height="wrap_content"
              android:gravity="center">
    <Button
        android:id="@+id/bnLogin"
        android:text="登录"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>
    <Button
        android:id="@+id/bnCancel"
        android:text="取消"
```

```
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"/>
    </LinearLayout>

</TableLayout>
```

- DialogUtil.java

```
package com.example.administrator.auctionclient.util;

import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.view.View;

import com.example.administrator.auctionclient.MainActivity;

/**
 * Created by Administrator on 2018-8-15 0015.
 */
public class DialogUtil {
    //定义一个显示消息 的对话框
    public static void showDialog(final Context ctx,String msg,boolea
n goHome)
    {
        //创建一个 AlertDialog.Builder 对象
        AlertDialog.Builder builder=new AlertDialog.Builder(ctx)
                .setMessage(msg).setCancelable(false);
        if (goHome)
        {
```

```
            builder.setPositiveButton("确定", new DialogInterface.OnC
lickListener() {
                @Override
                public void onClick(DialogInterface dialog, int which)
 {
                    //返回主页面
                    Intent intent=new Intent(ctx, MainActivity.clas
s);
                    intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                    ctx.startActivity(intent);


                }
            });
        }
        else {
            builder.setPositiveButton("确定",null);
        }
        builder.create().show();
    }
    //定义一个显示指定组件的对话框
    public static void showDialog(Context ctx, View view)
    {
        new AlertDialog.Builder(ctx).setView(view).setCancelable(fals
e)
                .setPositiveButton("确定",null)
                .create().show();
    }
}
```

- MainActivity.java

```
package com.example.administrator.auctionclient;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ListView;


public class MainActivity extends Activity {

    ListView auction_list;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        auction_list=(ListView)findViewById(R.id.auction_list);

        //为列表项的点击事件绑定事件监听器
        auction_list.setOnItemClickListener(new AdapterView.OnItemCli
ckListener() {
            @Override
            public void onItemClick(AdapterView<?> parent, View view,
 int position, long id) {
                Intent intent=null;
                switch ((int)id)
                {
```

```
                    case 0://查看竞得物品
                        //启动 viewItem Activity
                        intent=new Intent(MainActivity.this,ViewItem.
class);
                        intent.putExtra("action","viewSucc.jsp");
                        startActivity(intent);
                        break;
                    case 1://浏览流拍物品，启动 ViewItem Activity
                        intent=new Intent(MainActivity.this,ViewItem.
class);
                        intent.putExtra("action","viewFail.jsp");
                        startActivity(intent);
                        break;
                    case 2://管理物品种类，启动 managekind Activity
                        intent=new Intent(MainActivity.this,ManageKin
d.class);
                        startActivity(intent);
                        break;
                    case 3://管理物品，启动 manageItem Activity
                        intent=new Intent(MainActivity.this,ManageIte
m.class);
                        startActivity(intent);
                        break;
                    case 4://浏览拍卖物品，启动 ChooseKind Activity
                        intent=new Intent(MainActivity.this,ChooseKin
d.class);
                        startActivity(intent);
                        break;
                    case 5:
```

```
                        intent=new Intent(MainActivity.this,ViewBid.c
lass);
                        startActivity(intent);
                        break;

                    }
                }
            });


    }
}
```

- res/values/array.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string-array name="auction_list">
        <item>查看竞得物品</item>
        <item>浏览流拍物品</item>
        <item>管理物品种类</item>
        <item>管理物品</item>
        <item>浏览拍卖物品</item>
        <item>查看自己的竞标</item>
    </string-array>
    <string-array name="availTime">
        <item>一天</item>
        <item>二天</item>
        <item>三天</item>
        <item>四天</item>
        <item>五天</item>
```

```
        <item>一个星期</item>
        <item>一个月</item>
    </string-array>
</resources>
```

记得在清单文件中加入访问网络的权限：

```
<uses-permission android:name="android.permission.INTERNET"/>
```

运行效果：（记得先开启服务器）

4.2 进入主页面

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/andro
id"
              android:orientation="vertical"
              android:gravity="center"
              android:layout_width="match_parent"
              android:layout_height="match_parent">
<TextView
    android:text="欢迎使用电子拍卖系统"
    android:id="@+id/TextView"
    android:textSize="20dp"
    android:gravity="center"
    android:padding="30dp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"/>
    <ListView
        android:id="@+id/auction_list"
        android:entries="@array/auction_list"
        android:layout_width="match_parent"
        android:layout_height="match_parent"></ListView>
```

```
</LinearLayout>
```

在清单文件中注册：

```
<activity android:name=".Login">
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>
              <category android:name="android.intent.category.LAUNCHE
          R"/>
          </intent-filter>
</activity>
<activity android:name=".MainActivity"/>
<activity android:name=".ViewItem"/>
<activity android:name=".ManageKind"/>
<activity android:name=".AddKind"/>
<activity android:name=".ManageItem"/>
<activity android:name=".AddItem"/>
<activity android:name=".ChooseKind"/>
<activity android:name=".ChooseItem"/>
<activity android:name=".Addbid"/>
<activity android:name=".ViewBid"/>
```

第五章 实现主页面各个功能

5.1 浏览竞得/流拍物品

在服务器上写浏览流拍物品功能：

- ViewFailServlet.java

```
@WebServlet(urlPatterns="/android/viewFail.jsp")
public class ViewFailServlet extends BaseServlet
{
    public void service(HttpServletRequest request ,
        HttpServletResponse response)
        throws IOException , ServletException
    {
        // 获取业务逻辑组件
        AuctionManager auctionManager = (AuctionManager)getCtx().getBe
an("mgr");
        // 获取系统内所有流拍的物品
        List<ItemBean> items = auctionManager.getFailItems();
        JSONArray jsonArr= new JSONArray(items);
        response.setContentType("text/html; charset=GBK");
        response.getWriter().println(jsonArr.toString());  // ①
    }
}
```

- ViewItem.java

```
public class ViewItem extends Activity {
    Button bnHome;
    ListView succList;
    TextView viewTitle;
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.view_item);

        //获取界面上的控件
        bnHome=(Button)findViewById(R.id.bn_home);
        viewTitle=(TextView)findViewById(R.id.view_title);
        succList=(ListView)findViewById(R.id.succList);

        //为返回按钮的单击事件绑定事件监听器
        bnHome.setOnClickListener(new HomeListener(this));

        String action=getIntent().getStringExtra("action");
        //定义发送请求的URL
        String url= HttpUtil.BASE_URL+action;
        //如果是查看流拍物品，修改标题
        if (action.equals("viewFail.jsp"))
        {
            viewTitle.setText("查看流拍物品");
        }
        try{
            //向指定 url 发送请求，并把服务器响应转化成 jsonarray
对象
            JSONArray jsonArray=new JSONArray(HttpUtil.getReq
uest(url));
            //将 jsonarray 包装成 adapter
            JSONArrayAdapter adapter=new JSONArrayAdapter(thi
s,jsonArray,"name",true);
            succList.setAdapter(adapter);
```

```
        }catch (Exception e)
        {
            DialogUtil.showDialog(this,"服务器响应异常，请稍后再试！！",false);
            e.printStackTrace();
        }

        succList.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
                //查看指定物品的详细情况

                //加载 detail.xml 界面布局代表的视图
                View detailView=getLayoutInflater().inflate(R.layout.detail,null);
                //获取界面中文本框
                TextView itemName=(TextView)detailView.findViewById(R.id.itemName);
                TextView itemKind=(TextView)detailView.findViewById(R.id.itemKind);
                TextView maxPrice=(TextView)detailView.findViewById(R.id.maxPrice);
                TextView itemRemark=(TextView)detailView.findViewById(R.id.itemRemark);

                //获取被单击的列表项
                JSONObject jsonObject=(JSONObject)succList.ge
```

```
tAdapter().getItem(position);
                try {//通过文本框显示物品详情
                    itemName.setText(jsonObject.getString("na
me"));
                    itemKind.setText(jsonObject.getString("ki
nd"));
                    maxPrice.setText(jsonObject.getString("ma
xPrice"));
                    itemRemark.setText(jsonObject.getString("
desc"));


                }catch (Exception e)
                {
                    e.printStackTrace();
                }
                DialogUtil.showDialog(ViewItem.this,detailVie
w);
            }
        });
    }
}
```

HomeListener.java

```
package com.example.administrator.auctionclient;

import android.app.Activity;
import android.content.Intent;
import android.view.View;

/**
 * Created by Administrator on 2018-8-16 0016.
```

```
 */
public class HomeListener implements View.OnClickListener {
    private Activity activity;
    public HomeListener(Activity activity)
    {
        this.activity=activity;
    }
    @Override
    public void onClick(View v) {
        Intent intent=new Intent(activity,MainActivity.class);
        intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
        activity.startActivity(intent);
    }
}
```

- JSONArrayAdapter.java

```
public class JSONArrayAdapter extends BaseAdapter {
    private Context ctx;
    //定义需要包装的 JSONArray 对象
    private JSONArray jsonArray;
    //定义列表项显示 JSONObject 对象的哪个属性
    private  String property;
    private boolean haslcon;
    public JSONArrayAdapter(Context ctx,JSONArray jsonArray,String pr
operty,boolean haslcon)
    {
        this.ctx=ctx;
        this.jsonArray=jsonArray;
        this.property=property;
        this.haslcon=haslcon;


    }
    @Override
```

```
    public int getCount() {
        return jsonArray.length();
    }


    @Override
    public Object getItem(int position) {
        return jsonArray.optJSONObject(position);
    }

    @Override
    public long getItemId(int position) {
        try {
            return ((JSONObject)getItem(position)).getInt("id");
        }catch (Exception e)
        {
            e.printStackTrace();
        }
        return 0;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup par
ent) {
        //定义一个线性布局管理器
        LinearLayout linearLayout=new LinearLayout(ctx);
       //设置为水平的线性布局管理器
        linearLayout.setOrientation(LinearLayout.HORIZONTAL);
        //创建一个 ImageView
```

```
        ImageView imageView=new ImageView(ctx);
        imageView.setPadding(10,0,20,0);
        imageView.setImageResource(R.drawable.item);
        //将图片添加到 LinearLayout 中
        linearLayout.addView(imageView);
        //创建一个 textview
        TextView textView=new TextView(ctx);
        try {
            //获取 JSONArray 数组元素的 property 属性
            String itemName=((JSONObject)getItem(position)).getString
(property);
            //设置 textview 所显示的内容
            textView.setText(itemName);
        }catch (Exception e)
        {
            e.printStackTrace();
        }
        textView.setTextSize(20);
        if (hasIcon)
        {
            //将 textview 添加到 linearlayout 中
            linearLayout.addView(textView);
            return linearLayout;
        }
        else
        {
            return textView;
        }
```

```
    }
}
```

5.2 管理物品种类

ViewKindServlet.java

```
@WebServlet(urlPatterns="/android/viewKind.jsp")
public class ViewKindServlet extends BaseServlet
{
    public void service(HttpServletRequest request ,
        HttpServletResponse response)
        throws IOException , ServletException
    {
        // 获取业务逻辑组件
        AuctionManager auctionManager = (AuctionManager)getCtx().getBe
an("mgr");
        // 获取系统中所有物品种类
        List<KindBean> kinds = auctionManager.getAllKind();
        // 将所有物品种类包装成 JSONArray
        JSONArray jsonArr= new JSONArray(kinds);
        response.setContentType("text/html; charset=GBK");
        // 将 JSONArray 转换成 JSON 字符串后输出到客户端
        response.getWriter().println(jsonArr.toString());
    }
}
```

ManageKind.java

```
public class ManageKind extends Activity {
   Button bnHome,bnAdd;
    ListView kindList;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
        setContentView(R.layout.manage_kind);

        //获取界面布局上的按钮
        bnHome=(Button)findViewById(R.id.bn_home);
        bnAdd=(Button)findViewById(R.id.bnAdd);
        //为返回按钮的单击事件绑定事件监听器
        bnHome.setOnClickListener(new HomeListener(this));

        //获取界面布局上的列表
        kindList=(ListView)findViewById(R.id.kindList);

        //为列表添加数据
        //定义发送请求的 URL
        String url=HttpUtil.BASE_URL+"viewKind.jsp";
        try{//向指定 URL 发送请求，并把响应包装成 JSONArray 对象
            final JSONArray jsonArray=new JSONArray(HttpUtil.getReque
st(url));
            //把 JSONArray 对象包装成 adapter
            kindList.setAdapter(new KindArrayAdapter(this,jsonArra
y));

        }catch (Exception e)
        {
            DialogUtil.showDialog(ManageKind.this,"服务器响应异常，请
稍后再试！！",false);
            e.printStackTrace();
        }
```

```
        bnAdd.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent=new Intent(ManageKind.this,AddKind.clas
s);
                startActivity(intent);
            }
        });
    }
}
```

KindArrayAdapter.java

```
public class KindArrayAdapter extends BaseAdapter {
    private Context ctx;
    //定义需要包装的 JSONArray 对象
    private JSONArray kindjsonArray;

    public KindArrayAdapter(Context ctx, JSONArray jsonArray)
    {
        this.ctx=ctx;
        this.kindjsonArray=jsonArray;

    }
    @Override
    public int getCount() {
        return kindjsonArray.length();
    }


    @Override
```

```
    public Object getItem(int position) {
        return kindjsonArray.optJSONObject(position);
    }

    @Override
    public long getItemId(int position) {
        try {
            return ((JSONObject)getItem(position)).getInt("id");
        }catch (Exception e)
        {
            e.printStackTrace();
        }
        return -1;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup par
ent) {
       //定义一个线性布局管理器
        LinearLayout linearLayout=new LinearLayout(ctx);
        //设置为水平布局管理
        linearLayout.setOrientation(LinearLayout.VERTICAL);
        //定义一个线性布局管理器显示每个物品类
        LinearLayout temp=new LinearLayout(ctx);
        //设置为垂直布局管理
        temp.setOrientation(LinearLayout.HORIZONTAL);
        //创建一个 imageview
        ImageView imageView=new ImageView(ctx);
        imageView.setPadding(10,0,20,0);
```

```
        imageView.setImageResource(R.drawable.item);
        //将图片添加到 temp 布局管理器中
        temp.addView(imageView);

        //创建一个 Textview
        TextView textView=new TextView(ctx);
        try {
            //获取 jsonArray 数组元素中的 kindname 属性
            String kindname=((JSONObject)getItem(position)).getString
("kindName");
            textView.setText(kindname);
        }catch (Exception e)
        {
            e.printStackTrace();
        }
        textView.setTextSize(20);
        //将 textview 添加到 temp 布局管理器中
        temp.addView(textView);
        linearLayout.addView(temp);

        //定义一个文本框来显示种类描述
        TextView desc=new TextView(ctx);
        desc.setPadding(10,0,20,0);
        try
        {
            //获取 jsonarray 数组元素的 kinddesc 属性
            String kindDesc=((JSONObject)getItem(position)).getString
("kindDesc");
            desc.setText(kindDesc);
```

```
        }catch (Exception e)
        {
            e.printStackTrace();
        }
        desc.setTextSize(20);
        linearLayout.addView(desc);

        return linearLayout;
    }
}
```

AddKindServlet.java

```
@WebServlet(urlPatterns="/android/addKind.jsp")
public class AddKindServlet extends BaseServlet
{
    public void service(HttpServletRequest request ,
        HttpServletResponse response)
        throws IOException , ServletException
    {
        request.setCharacterEncoding("gbk");
        // 获取请求参数
        String name = request.getParameter("kindName");
        String desc = request.getParameter("kindDesc");
        // 获取系统业务逻辑组件
        AuctionManager auctionManager = (AuctionManager)getCtx().getBe
an("mgr");
        // 调用业务逻辑组件的业务方法添加种类
        int kindId = auctionManager.addKind(new Kind(name , desc));
        response.setContentType("text/html; charset=GBK");
        // 添加成功
```

```
        if (kindId > 0)
        {
            response.getWriter().println("恭喜您, 种类添加成功!");
        }
        else
        {
            response.getWriter().println("对不起, 种类添加失败!");
        }
    }
}
```

AddKind.java

```
public class AddKind extends Activity {
    //定义界面中的控件
    Button bnAdd,bnCancel;
    EditText kindName,kindDesc;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.add_kind);
        //获取布局界面中的控件
        bnAdd=(Button)findViewById(R.id.bnAdd);
        bnCancel=(Button)findViewById(R.id.bnCancel);
        kindName=(EditText)findViewById(R.id.kindName);
        kindDesc=(EditText)findViewById(R.id.kindDesc);

        //为取消按钮的单击事件绑定事件监听器
        bnCancel.setOnClickListener(new HomeListener(this));

        //为添加按钮的单击事件绑定事件监听器
```

```
        bnAdd.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View v) {
                //首先校验文本框的内容不为空,然后把物品种类和描述添加到数据库中
                if (validate())
                {
                    //获取文本框中的种类名称和描述
                    String name=kindName.getText().toString();
                    String desc=kindDesc.getText().toString();
                    //添加物品种类
                    try{
                        //使用 map 封装请求参数
                        Map<String,String> map=new HashMap<String, String>();
                        map.put("kindName",name);
                        map.put("kindDesc",desc);
                        //定义发送请求的 URL
                        String url=HttpUtil.BASE_URL+"addKind.jsp";
                        String result=HttpUtil.postRequest(url,map);
                        //使用对话框来显示添加结果
                        DialogUtil.showDialog(AddKind.this,result,true);

                    }catch (Exception e){
                        DialogUtil.showDialog(AddKind.this,"服务器响应异常, 请稍后再试!!",false);
                        e.printStackTrace();
                    }
```

```
                }
            }
        });
    }
    private boolean validate()
    {
        String name=kindName.getText().toString().trim();
        if (name.equals(""))
        {
            DialogUtil.showDialog(this,"种类名称不能为空！！",false);
            return false;
        }
        String desc=kindDesc.getText().toString().trim();
        if (desc.equals(""))
        {
            DialogUtil.showDialog(this,"种类描述不能为空！！",false);
            return false;
        }
        return true;
    }
}
```

5.3 管理物品

ViewOwnerItemServlet.java

```
@WebServlet(urlPatterns="/android/viewOwnerItem.jsp")
public class ViewOwnerItemServlet extends BaseServlet
{
    public void service(HttpServletRequest request ,
        HttpServletResponse response)
        throws IOException , ServletException
    {
        // 获取 userId
        Integer userId = (Integer)request.getSession(true)
            .getAttribute("userId");
        // 获取业务逻辑组件
        AuctionManager auctionManager = (AuctionManager)getCtx().getBe
an("mgr");
```

```
        // 获取该用户当前处于拍卖中的所有物品
        List<ItemBean> items = auctionManager.getItemsByOwner(userId);
        // 将查询得到的物品封装成 JSONArray 对象
        JSONArray jsonArr= new JSONArray(items);
        response.setContentType("text/html; charset=GBK");
        response.getWriter().println(jsonArr.toString());
    }
}
```

ManageItem.java

```
public class ManageItem extends Activity {
    Button bnHome,bnAdd;
     ListView itemList;
     @Override
     protected void onCreate(Bundle savedInstanceState) {
         super.onCreate(savedInstanceState);
         setContentView(R.layout.manage_item);

         //获取界面布局上的按钮
         bnHome=(Button)findViewById(R.id.bn_home);
         bnAdd=(Button)findViewById(R.id.bnAdd);
         //为返回按钮的单击事件绑定事件监听器
         bnHome.setOnClickListener(new HomeListener(this));

         //获取界面布局上的列表
         itemList=(ListView)findViewById(R.id.itemlist);

         //为列表添加数据
         //定义发送请求的 URL
         String url=HttpUtil.BASE_URL+"viewOwnerItem.jsp";
```

```
        try{//向指定 URL 发送请求，并把响应包装成 JSONArray 对象
            final JSONArray jsonArray=new JSONArray(HttpUtil.getReque
st(url));
            //把 JSONArray 对象包装成 adapter

            itemList.setAdapter(new JSONArrayAdapter(this,jsonArray,"
name",true));

        }catch (Exception e)
        {
            DialogUtil.showDialog(ManageItem.this,"服务器响应异常，请
稍后再试！！",false);
            e.printStackTrace();
        }
        //单击列表项时显示详细信息
        itemList.setOnItemClickListener(new OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> parent, View view,
 int position, long id) {
                //加载 detail_in_bid.xml 布局代表的视图
                View detailView=getLayoutInflater().inflate(R.layout.
detail_in_bid,null );
                //获取 detail_in_bid.xml 界面中的文本框
                TextView itemName=(TextView)detailView.findViewById(R.
id.itemName);
                TextView itemKind=(TextView)detailView.findViewById(R.
id.itemKind);
                TextView maxPrice=(TextView)detailView.findViewById(R.
id.maxPrice);
```

```
                TextView initPrice=(TextView)detailView.findViewById
(R.id.initPrice);
                TextView endTime=(TextView)detailView.findViewById(R.
id.endTime);
                TextView itemRemark=(TextView)detailView.findViewById
(R.id.itemRemark);

                //获取单击列表项所包装的 jsonobject
                JSONObject jsonObject=(JSONObject)itemList.getAdapter
().getItem(position);
                //通过文本框显示物品详情
                try
                {
                    itemName.setText(jsonObject.getString("name"));
                    itemKind.setText(jsonObject.getString("kind"));
                    maxPrice.setText(jsonObject.getString("maxPrice
"));
                    itemRemark.setText(jsonObject.getString("desc"));
                    initPrice.setText(jsonObject.getString("initPrice
"));
                    endTime.setText(jsonObject.getString("endTime"));
                }catch (Exception e)
                {
                    e.printStackTrace();
                }
                DialogUtil.showDialog(ManageItem.this,detailView);
            }
        });
```

```
        bnAdd.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent=new Intent(ManageItem.this,AddItem.clas
s);
                startActivity(intent);
            }
        });
    }
}
```

AddItemServlet.java

```
@WebServlet(urlPatterns="/android/addItem.jsp")
public class AddItemServlet extends BaseServlet
{
    public void service(HttpServletRequest request ,
       HttpServletResponse response)
       throws IOException , ServletException
    {
       // 获取 userId
       Integer userId = (Integer)request.getSession(true)
          .getAttribute("userId");
       request.setCharacterEncoding("gbk");
       // 解析请求参数
       String itemName = request.getParameter("itemName");
       String itemDesc = request.getParameter("itemDesc");
       String remark = request.getParameter("itemRemark");
       String initPrice = request.getParameter("initPrice");
       String kindId = request.getParameter("kindId");
       String avail = request.getParameter("availTime");
```

```
        // 获取业务逻辑组件
        AuctionManager auctionManager = (AuctionManager)getCtx().getBe
an("mgr");
        // 调用业务逻辑组件的方法来添加物品
        int itemId = auctionManager.addItem(new Item(itemName , itemDe
sc
            , remark , Double.parseDouble(initPrice))
            , Integer.parseInt(avail) , Integer.parseInt(kindId) , use
rId);
        response.setContentType("text/html; charset=GBK");
        // 添加成功
        if (itemId > 0)
        {
            response.getWriter().println("恭喜您, 物品添加成功!");
        }
        else
        {
            response.getWriter().println("对不起, 物品添加失败!");
        }
    }
}
```

AddItem.java

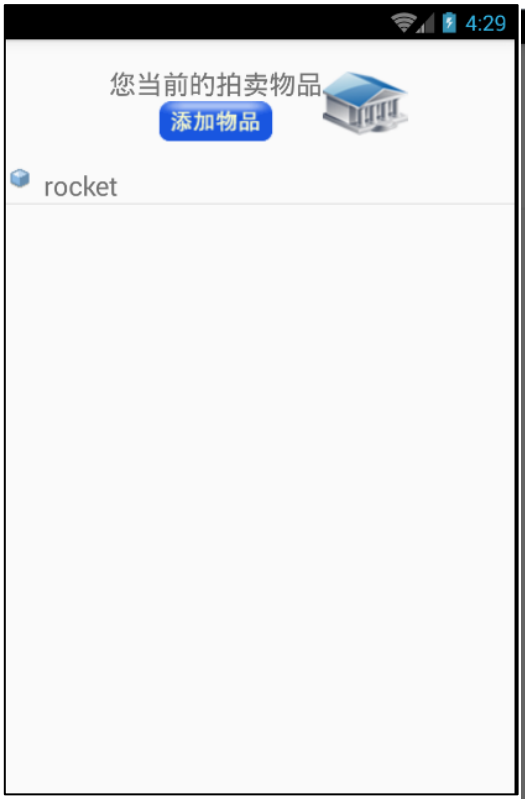
```
public class AddItem extends Activity {
    //定义界面中的文本框
    EditText itemName,itemDesc,itemRemark,initPrice;
    Spinner itemKind,availTime;
    //定义界面中两个按钮
    Button bnAdd,bnCancel;
    @Override
```

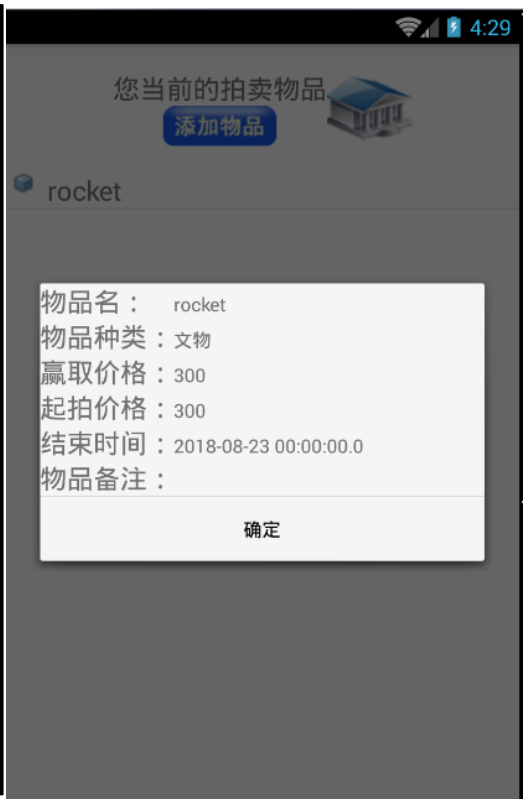
```
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.add_item);

        //获取界面中的文本框
        itemName=(EditText)findViewById(R.id.itemName);
        itemDesc=(EditText)findViewById(R.id.itemDesc);
        itemRemark=(EditText)findViewById(R.id.itemRemark);
        initPrice=(EditText)findViewById(R.id.initPrice);
        itemKind=(Spinner)findViewById(R.id.itemKind);
        availTime=(Spinner)findViewById(R.id.availTime);

        //定义发送请求的地址
        String url=HttpUtil.BASE_URL+"viewKind.jsp";
        JSONArray jsonArray=null;
        try
        {//获取系统中所有的物品种类
            //向执行 URL 发送请求，并把服务器响应转换成 jsonarray
            jsonArray=new JSONArray(HttpUtil.getRequest(url));


        }catch (Exception e)
        {
            e.printStackTrace();
        }
        //将 jsonarray 包装成 adapter
        JSONArrayAdapter jsonArrayAdapter=new JSONArrayAdapter(this,j
sonArray,"kindName",false);
        //显示物品种类列表
```

```
        itemKind.setAdapter(jsonArrayAdapter);
        //获取界面中的两个按钮
        bnAdd=(Button)findViewById(R.id.bnAdd);
        bnCancel=(Button)findViewById(R.id.bnCancel);
        bnCancel.setOnClickListener(new HomeListener(this));
        bnAdd.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View v) {
                //首先对输入的内容进行校验，然后在添加到数据库中
                if (validate())
                {
                    //获取用户输入的物品名称/物品描述等信息
                    String name=itemName.getText().toString();
                    String desc=itemDesc.getText().toString();
                    String remark=itemRemark.getText().toString();
                    String price=initPrice.getText().toString();
                    JSONObject kind=(JSONObject) itemKind.getSelected
Item();
                    int avail=availTime.getSelectedItemPosition();
                    //根据哟韩国胡选择的有效时间选项,指定实际的有效时
间
                    switch (avail)
                    {
                        case 5:avail=7;break;
                        case 6:avail=30;break;
                        default:avail+=1;break;
                    }
                    //添加物品
                    try
```

```
                    {
                        //使用 map 封装请求参数
                        Map<String ,String > map=new HashMap<String,
String>();
                        map.put("itemName",name);
                        map.put("itemDesc",desc);
                        map.put("itemRemark",remark);
                        map.put("initPrice",price);
                        map.put("kindId",kind.getInt("id")+"");
                        map.put("availTime",avail+"");
                        //定义发送请求的 URL
                        String url=HttpUtil.BASE_URL+"addItem.jsp";
                        // 发送请求
                        String result=HttpUtil.postRequest(url,map);
                        // 显示对话框
                        DialogUtil.showDialog(AddItem.this,result,tru
e);
                    }catch (Exception e)
                    {
                        DialogUtil.showDialog(AddItem.this,"服务器响
应异常，请稍后再试！！",false);
                        e.printStackTrace();
                    }
                }
            }
        });
    }
    private boolean validate()
    {
```

```
        String name=itemName.getText().toString().trim();
        if (name.equals(""))
        {
            DialogUtil.showDialog(this,"物品名称是必须要填写的！！",f
alse);
            return false;
        }
        String price=initPrice.getText().toString().trim();
        if (price.equals(""))
        {
            DialogUtil.showDialog(this,"起拍价格是必须要填写的！！",f
alse);
            return false;
        }
        //尝试将起拍价格转换为浮点数
        try
        {
            Double.parseDouble(price);
        }catch (Exception e)
        {
            DialogUtil.showDialog(this,"起拍价格必须是数值!!",false);
            e.printStackTrace();
        }
        return true;}
```


4:29
您当前的拍卖物品
添加物品
rocket


4:29
您当前的拍卖物品
添加物品
rocket
物品名： rocket
物品种类：文物
赢取价格：300
起拍价格：300
结束时间：2018-08-23 00:00:00.0
物品备注：
确定


4:29
添加拍卖物品
物品名：
物品描述：
物品备注：
起拍价格：
有效时间： 一天
物品种类： 电脑硬件
添加
取消

5.4 浏览拍卖物品

ChooseKind.class

```
public class ChooseKind extends Activity {
    Button bnHome;
    ListView kindList;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.choose_kind);
        //获取界面布局中的控件
        bnHome=(Button)findViewById(R.id.bn_home);
        kindList=(ListView)findViewById(R.id.kindList);
        bnHome.setOnClickListener(new HomeListener(this));
        //定义发送请求的Url
        String url=HttpUtil.BASE_URL+"viewKind.jsp";
        try
        {
            //向指定url发送请求，并将服务器响应包装jsonarray对象
            JSONArray jsonArray=new JSONArray(HttpUtil.getRequest(ur
l));
            //使用kindlist显示所有物品
            kindList.setAdapter(new KindArrayAdapter(ChooseKind.this,
jsonArray));
        }catch (Exception e)
        {
            DialogUtil.showDialog(this,"服务器响应异常，请稍后再试！！！
",false);
            e.printStackTrace();
        }
```

```
        kindList.setOnItemClickListener(new OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> parent, View view,
int position, long id) {
                Intent intent=new Intent(ChooseKind.this,ChooseItem.cl
ass);
                intent.putExtra("kindId",id);
                startActivity(intent);
            }
        });
    }
}
```

chooseItem.java

```
public class ChooseItem extends Activity {
    Button bnHome;
    ListView succList;
    TextView viewTitle;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.view_item);
        //获取界面中的控件
        bnHome=(Button)findViewById(R.id.bn_home);
        succList=(ListView)findViewById(R.id.succList);
        viewTitle=(TextView)findViewById(R.id.view_title);
        bnHome.setOnClickListener(new HomeListener(this));
        viewTitle.setText("当前种类的物品");
        //获取传递过来的种类 id
        long id=getIntent().getLongExtra("kindId",0);
```

```
        // 定义发送请求的URI
        String url=HttpUtil.BASE_URL+"itemList.jsp?kindId="+id;
        try
        {
            //根据种类id获取该种类所对应的所有物品
            JSONArray jsonArray=new JSONArray(HttpUtil.getRequest(ur
l));
            JSONArrayAdapter jsonArrayAdapter=new JSONArrayAdapter(th
is,jsonArray,"name",true);
            //使用listview显示当前种类的所有物品
            succList.setAdapter(jsonArrayAdapter);
        }catch (Exception e)
        {
            DialogUtil.showDialog(ChooseItem.this,"服务器响应异常，请
稍后再试！！",false);
            e.printStackTrace();
        }

        succList.setOnItemClickListener(new OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> parent, View view,
 int position, long id) {
                try{
                    JSONObject jsonObject=(JSONObject) succList.getAd
apter().getItem(position);
                    Intent intent=new Intent(ChooseItem.this,Addbid.c
lass);
                    intent.putExtra("itemId",jsonObject.getInt("id
"));
```

```
                    // Toast.makeText(ChooseItem.this, jsonObject.getI
nt("id")+"", Toast.LENGTH_SHORT).show();
                        startActivity(intent);
                    }catch (Exception e)
                    {
                        e.printStackTrace();
                    }
                }
            });
        }
}
AddBidServlet.java
@WebServlet(urlPatterns="/android/addBid.jsp")
public class AddBidServlet extends BaseServlet
{
    public void service(HttpServletRequest request ,
        HttpServletResponse response)
        throws IOException , ServletException
    {
        // 获取 userId
        Integer userId = (Integer)request.getSession(true)
            .getAttribute("userId");
        request.setCharacterEncoding("gbk");
        // 获取请求参数
        String itemId = request.getParameter("itemId");
        String bidPrice = request.getParameter("bidPrice");
        // 获取业务逻辑组件
        AuctionManager auctionManager = (AuctionManager)getCtx().getBe
an("mgr");
```

```
        // 调用业务方法来添加竞价
        int bidId = auctionManager.addBid(Integer.parseInt(itemId)
            , new Bid(Double.parseDouble(bidPrice))
            , userId);
        response.setContentType("text/html; charset=GBK");
        // 竞价成功
        if (bidId >= 0)
        {
            response.getWriter().println("恭喜您, 竞价成功!");
        }
        else
        {
            response.getWriter().println("对不起, 竞价失败!");
        }
    }
}
```

Addbid.class

```
public class Addbid extends Activity {
    //定义界面中的文本框
    TextView itemName,itemDesc,itemRemark,itemKind,initPrice,maxPrice,
endTime;
    EditText bidPrice;
    Button bnAdd,bnCancel;
    //定义当前正在拍卖的物品
    JSONObject jsonObject;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.add_bid);
```

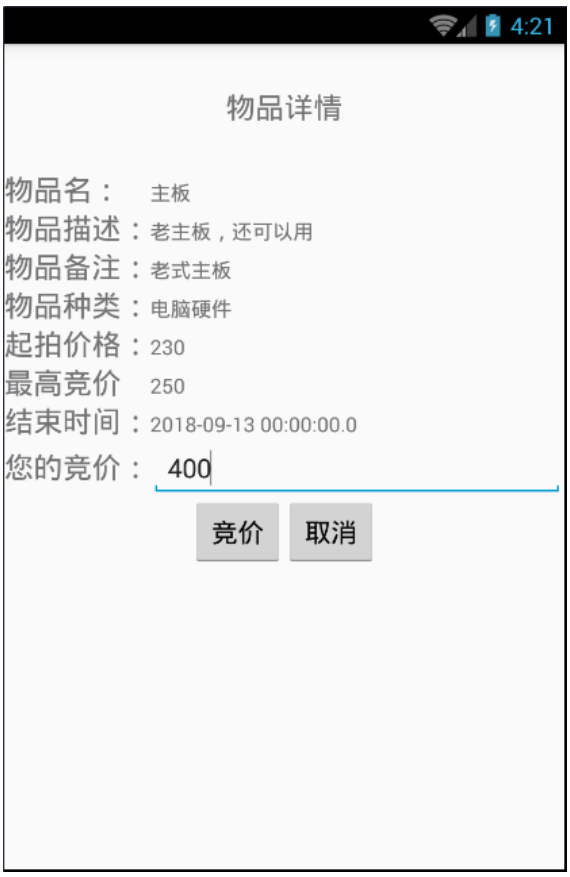
```
//获取界面中的编辑框
itemName=(TextView)findViewById(R.id.itemName);
itemDesc=(TextView)findViewById(R.id.itemDesc);
itemRemark=(TextView)findViewById(R.id.itemRemark);
itemKind=(TextView)findViewById(R.id.itemKind);
initPrice=(TextView)findViewById(R.id.initPrice);
maxPrice=(TextView)findViewById(R.id.maxPrice);
endTime=(TextView)findViewById(R.id.endTime);
bidPrice=(EditText)findViewById(R.id.bidPrice);
bnAdd=(Button)findViewById(R.id.bnAdd);
bnCancel=(Button)findViewById(R.id.bnCancel);
bnCancel.setOnClickListener(new HomeListener(this));
//获取指定拍卖物品的 id
final int itemid=getIntent().getIntExtra("itemId",0);
//定义发送请求的 Url
String url=HttpUtil.BASE_URL+"getItem.jsp?itemId="+itemid;
try {
    //获取指定的拍卖物品
    jsonObject=new JSONObject(HttpUtil.getRequest(url));
    //使用文本框显示拍卖物品的详情
    itemName.setText(jsonObject.getString("name"));
    itemDesc.setText(jsonObject.getString("desc"));
    itemRemark.setText(jsonObject.getString("remark"));
    itemKind.setText(jsonObject.getString("kind"));
    initPrice.setText(jsonObject.getString("initPrice"));
    maxPrice.setText(jsonObject.getString("maxPrice"));
    endTime.setText(jsonObject.getString("endTime"));

}catch (Exception e)
```

```
        {
            DialogUtil.showDialog(Addbid.this,"服务器出现异常，请稍候再试！！",false);
            e.printStackTrace();
        }
        bnAdd.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View v) {
                try
                {
                    //执行类型转化
                    double curPrice=Double.parseDouble(bidPrice.getText().toString());
                    //执行校验
                    if (curPrice<jsonObject.getDouble("maxPrice"))
                    {
                        DialogUtil.showDialog(Addbid.this,"您输入的竞价必须高于当前竞价",false);
                    }
                    else
                    {
                        //添加竞价记录
                        //使用 map 封装请求参数
                        Map<String,String> map=new HashMap<String, String>();
                        map.put("itemId",jsonObject.getString("id"));
                        map.put("bidPrice",curPrice+"");
                        //定义请求的 URl
                        String url=HttpUtil.BASE_URL+"addBid.jsp";
```

```
                        //发送请求
                        String result=HttpUtil.postRequest(url,map);
                        //显示对话框提示竞价成功
                        DialogUtil.showDialog(Addbid.this,result,tru
e);
                    }
                }catch (NumberFormatException ne)
                {
                    DialogUtil.showDialog(Addbid.this,"您输入的竞价必
须是数值",false);
                    ne.printStackTrace();
                }catch (Exception e)
                {
                    DialogUtil.showDialog(Addbid.this,"服务器响应异
常，请稍后再试！！",false);
                    e.printStackTrace();
                }
            }
        });
    }
}
```


4:21
请选择一个物品种类
电脑硬件
这里并不是很主流的产品，但价格绝对令你心动
房产
提供非常稀缺的房源
文物
这是一件来自明朝的瓷器
飞行器
缩短你的飞行距离


4:21
当前种类的物品
主板


4:21
物品详情
物品名： 主板
物品描述：老主板，还可以用
物品备注：老式主板
物品种类：电脑硬件
起拍价格：230
最高竞价 250
结束时间：2018-09-13 00:00:00.0
您的竞价： 400
竞价
取消

5.5 查看自己的竞标

ViewBid.java

```
public class ViewBid extends Activity {
    Button bnHome;
    ListView bidList;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.view_bid);
        //获取界面中的控件
        bnHome=(Button)findViewById(R.id.bn_home);
        bidList=(ListView)findViewById(R.id.bidList);
         bnHome.setOnClickListener(new HomeListener(this));
        //定义发送请求的URL
        String url=HttpUtil.BASE_URL+"viewBid.jsp";
        try
        {
            JSONArray jsonArray=new JSONArray(HttpUtil.getRequest(ur
l));
            JSONArrayAdapter jsonArrayAdapter=new JSONArrayAdapter(th
is,jsonArray,"item",true);
            bidList.setAdapter(jsonArrayAdapter);
        }catch (Exception e)
        {
            DialogUtil.showDialog(ViewBid.this,"服务器异常，请稍后再
试！！",false);
            e.printStackTrace();
        }
```

```
        bidList.setOnItemClickListener(new OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> parent, View view,
 int position, long id) {
                //加载 bid_detail 界面布局代表的视图
                View detailView=getLayoutInflater().inflate(R.layout.
bid_detail,null);
                //获取界面中的控件
                TextView itemName=(TextView)detailView.findViewById(R.
id.itemName);
                TextView bidPrice=(TextView)detailView.findViewById(R.
id.bidPrice);
                TextView bidTime=(TextView)detailView.findViewById(R.
id.bidTime);
                TextView bidUser=(TextView)detailView.findViewById(R.
id.bidUser);

                // 获取被单击项目所包装的 jsonobject
                JSONObject jsonObject=(JSONObject) bidList.getAdapter
().getItem(position);
                //使用文本框显示竞价详情
                try{
                    itemName.setText(jsonObject.getString("item"));
                    bidPrice.setText(jsonObject.getString("price"));
                    bidTime.setText(jsonObject.getString("bidDate"));
                    bidUser.setText(jsonObject.getString("user"));
                }catch (Exception e)
                {
                    e.printStackTrace();
```

```
                }
                DialogUtil.showDialog(ViewBid.this,detailView);
            }
        });
    }
}
```