# 交换说明

* 1. 交换格式

整个App中客户端与服务器通过Json进行应用层数据通信。

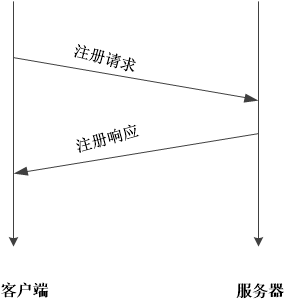
* + 1. 命名规范

1. Key的命名采用小驼峰命名法，如studentNo，phone
   1. 文档说明

当交互过程有更改时，应在双方协商好后及时更新此文档。

# 注册

* 1. 交互图



* 1. 注册请求

1. 方式：Post
2. 服务器URL：/user/register/
3. body内容

|  |  |
| --- | --- |
| 键（key） | 说明 |
| Name |  |
| Passwd |  |
| Phone |  |
| school |  |
| studentNo | 学号 |
| Email |  |
| gender | 性别，两个可能值。female,male |

* 1. 注册响应

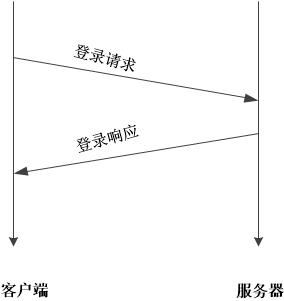
服务器根据用户名唯一性检查合法性。

1. 内容

|  |  |
| --- | --- |
| 键（key） | 说明 |
| Status | 0，注册成功  1，已经注册  2，无效参数 |
| Data | 0，用户完整信息  1，this username has been registered  2，invalid parameters |

# 登录

* 1. 交互图



* 1. 登录请求

1. 方式：Post
2. 服务器URL：/user/login/
3. body内容

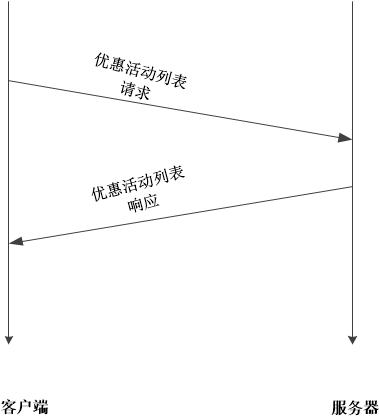
|  |  |
| --- | --- |
| 键（key） | 说明 |
| Name |  |
| passwd |  |

* 1. 登录响应

内容

|  |  |
| --- | --- |
| 键（key） | 说明 |
| Status | 0，登录成功  1，登录无效  2，无效参数 |
| Data | 0，用户完整信息  1，invalid username or pwssword  2，invalid parameters |

# 优惠活动

* 1. 交互图
  2. 优惠活动列表请求

1. 方式：Get
2. 服务器URL：/promotion/list/
3. body

无

* 1. 优惠活动列表响应

服务器返回Json数组形式的活动列表，列表中每个数据项的内容如下

|  |  |
| --- | --- |
| 键（key） | 说明 |
| promotionCompany | 活动电商名称 |
| promotionDeadline | 活动截止时间(注，有的活动没有截止时间) |
| promotionID | 参考《服务器设计补充》 |
| promotionName | 促销活动名称 |
| promotionLink | 促销活动在电商网站的首页链接url |

# 优惠活动中的搜索

* 1. 书籍搜索请求

客户端上，用户在某个优惠活动中根据书名搜索某本书籍。

1. 方式：get
2. 服务器URL1：/search/promotion/list/
3. 服务器URL2：/search/promotion/detail/
4. body1

|  |  |
| --- | --- |
| 键（key） | 说明 |
| promotionID | 促销活动ID |
| bookName | 目标书名 |

1. body2

|  |  |
| --- | --- |
| 键（key） | 说明 |
| promotionBookDetailLink | 活动图书详情页 |

* 1. 书籍搜索响应
     1. 书籍详细信息表

此表为URL1返回结果：

|  |  |
| --- | --- |
| 键（key） | 说明 |
| promotionBookISBN | 书籍ISBN |
| promotionBookName | 书名 |
|  |  |
| promotionBookImageLink | 书籍图片链接 |
|  |  |
|  |  |
|  |  |
| promotionBookPrice | 当前价格 |
|  |  |
|  |  |
|  |  |
| promotionBookDetailLink | 促销图书详情页链接（请求图书价格列表时的body内容） |

* + 1. 书籍比价信息表

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

此表存储某本书在各电商网站的价格信息（URL2返回的结果）。

|  |  |
| --- | --- |
| 键（key） | 说明 |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| bookISBN | 图书ISBN |
| bookSaler | 电商 |
| bookCurrentPrice | 相应电商的当前价 |
| bookLink | 活动图书链接 |

# 优惠活动详情

* 1. 优惠活动详情请求

1. 方式：get
2. 服务器URL：/promotion/detail/
3. body

|  |  |
| --- | --- |
| 键（key） | 说明 |
| promotionID | 促销活动ID |

* 1. 优惠活动详情响应

服务器返回Json数组形式的某活动的书籍列表（暂定20本，若用户有关注的书籍类别/书籍，则返回参加活动的此种类别、书籍。否则随机返回），列表中每个数据项的内容为书籍信息表。（参见6.2.1，此处不再赘述）

# 主页中的搜索

* 1. 搜索请求

用户可在主页进行书名搜索，查找指定书籍的相关活动和价格信息

1. 方式：get
2. 服务器URL1(请求结果列表)：/search/home/list/
3. 服务器URL2(请求结果列表中具体项的详细信息)
4. body1

|  |  |
| --- | --- |
| 键（key） | 说明 |
| bookName | 目标书名 |

1. body2

|  |  |
| --- | --- |
| 键（key） | 说明 |
| booSubject | 图书在豆瓣上的链接 |

* 1. 搜索响应

URL1返回内容如下：

|  |  |
| --- | --- |
| bookName | 书名 |
| booSubject | 图书在豆瓣上的链接 |
| bookImageLink | 书图片链接 |
| bookDetail | 书的简单信息 |
| bookLowestPrice | 书的最低价格 |

URL1返回内容实例：

[{"bookLowestPrice": "纸质版 72.40 元起", "bookName": "Java编程思想 （第4版）", "bookImageLink": "https://img3.doubanio.com/mpic/s27243455.jpg", "bookDetail": "[美] Bruce Eckel / 陈昊鹏 / 机械工业出版社 / 2007-6 / 108.00元", "booSubject": "https://book.douban.com/subject/2130190/"}, {"bookLowestPrice": "纸质版 198.00 元起", "bookName": "Java编程思想(第2版)", "bookImageLink": "https://img1.doubanio.com/mpic/s1085058.jpg", "bookDetail": "[美] Bruce Eckel / 侯捷 / 机械工业出版社 / 2002-9 / 99.00元", "booSubject": "https://book.douban.com/subject/1101158/"}]

URL2返回内容如下：

|  |  |
| --- | --- |
| bookISBN | 图书ISBN |
| bookSaler | 电商图片（非链接） |
| bookCurrentPrice | 当前价 |

URL2返回内容实例：

[{"bookCurrentPrice": "72.40", "bookISBN": "9787111213826", "bookSaler": "jingdong.png"}, {"bookCurrentPrice": "70.20", "bookISBN": "9787111213826", "bookSaler": "dangdang.png"}, {"bookCurrentPrice": "77.80", "bookISBN": "9787111213826", "bookSaler": "wenxuan.gif"}, {"bookCurrentPrice": "78.80", "bookISBN": "9787111213826", "bookSaler": "beifa.gif"}, {"bookCurrentPrice": "70.20", "bookISBN": "9787111213826", "bookSaler": "joyo.gif"}, {"bookCurrentPrice": "82.10", "bookISBN": "9787111213826", "bookSaler": "bookschina.jpg"}, {"bookCurrentPrice": "75.60", "bookISBN": "9787111213826", "bookSaler": "chinapub.jpg"}, {"bookCurrentPrice": "121.00", "bookISBN": "9787111213826", "bookSaler": "lanquan.jpg"}]

# 收藏

* 1. 添加收藏类别
     1. 添加收藏类别请求

1. 方式：Post
2. 服务器URL：/favorite/category/add/
3. body

|  |  |
| --- | --- |
| 键（key） | 说明 |
| Name | 用户名 |
| categoryList | 类别列表 |

* + - 1. 类别列表

类别列表中是用户请求添加的类别（表长>=1），Json数组，包含了若干个类别名。

* + 1. 添加收藏类别响应

指示添加类别的请求处理结果，内容待定

* 1. 删除收藏类别
     1. 删除收藏类别请求

1. 方式：Post
2. 服务器URL：/favorite/category/remove/
3. body

|  |  |
| --- | --- |
| 键（key） | 说明 |
| Name | 用户名 |
| categoryList | 请求删除的类别列表 |

* + 1. 删除收藏类别响应

指示添加收藏书籍的请求处理结果，内容待定

* 1. 添加收藏书籍
     1. 添加收藏书籍请求

1. 方式：Post
2. 服务器URL：/favorite /add/
3. body

|  |  |
| --- | --- |
| 键（key） | 说明 |
| userName | 用户名（暂未处理） |
| ISBN | 书籍列表 |
| bookName | 书籍名字（可为空） |

* + 1. 添加收藏书籍响应

|  |  |
| --- | --- |
| 键（key） | 说明 |
| Status | 0，请求成功  1，用户未登录  2，无效参数 |
| Data | 0，add success  1，user need login  2，invalid parameters |

* 1. 删除收藏书籍
     1. 删除收藏书籍请求

1. 方式：Post
2. 服务器URL：/favorite/remove/
3. body

|  |  |
| --- | --- |
| 键（key） | 说明 |
| userName | 用户名（暂未处理） |
| bookList | 书籍列表 |

* + 1. 删除收藏书籍响应

|  |  |
| --- | --- |
| 键（key） | 说明 |
| status | 0，请求成功  1，用户未登录  2，无效参数 |
| data | 0，delete success  1，user need login  2，invalid parameters |

* 1. 获取收藏列表
     1. 获取收藏书籍请求

1. 方式：GET
2. 服务器URL：/favorite/list/
   * 1. 获取收藏书籍响应

|  |  |
| --- | --- |
| 键（key） | 说明 |
| Status | 0，请求成功  1，用户未登录  2，无效参数 |
| Data | 0，{username: , favourite\_list: }  1，user need login  2，invalid parameters |

-----------------------------------------------------------------------------------------------------

# 拼购

* 1. 添加拼购书籍
     1. 添加拼购书籍请求

1. 请求格式：POST /order/add/
2. body

|  |  |
| --- | --- |
| 键（key） | 说明 |
| userName | 用户名 |
| promotionID | 促销活动ID |
| bookISBN | 书籍ISBN |

* + 1. 添加拼购书籍响应
  1. 获取拼购列表
     1. 请求

1. 请求格式：GET /order /list/
2. 请求body：

|  |  |
| --- | --- |
| 键（key） | 说明 |
| userName | 用户名 |
| promotionID | 促销活动ID |
| bookISBN | 书籍ISBN |

* + 1. 响应
  1. 确认拼购
     1. 请求

1. 请求格式：POST /order/submit
2. 请求body：
   * 1. 响应