**C/S通信协议设计**

1. **基础格式**

Fetcher带哥，服务器与客户端之间通过String进行通信，采用标准Json格式，每一个String包含一个JSONObject。定义如下：

1. **JSONObject**

JSONObject记录一个无序的“‘name/value’对”集合。一个JSONObject以“{”（左括号）开始，“}”（右括号）结束。每个“名称”后跟一个“:”（冒号）；“‘name/value’ 对”之间使用“,”（逗号）分隔。

**示例：**{“UserName”:”王帆”,”Nickname”:”Raven”,”age”:20 }

1. **name, value定义**

name必须为带双引号的String，一般用以说明value的类别。

value可以是双引号括起来的String、number、true、false、 null、JSONObject或者JSONArray。这些结构可以嵌套。

1. **JSONArray**

JSONArray是value的有序集合。一个JSONArray以“[”（左中括号）开始，“]”（右中括号）结束。value之间使用“,”（逗号）分隔。

**示例：**[1,”wangfan”,true,{“age”:20}]

1. **协议组织**
2. **基础JSONObject组织**

服务器与客户端之间每次传输包含一个JSONObject的字符串，组织如下：

{“type”:intvalue,”state”:intvalue,”data”:value}。其中，”type”:value“‘name/value’对”表示请求的类型，相同的值在客户端、服务器端有不同解释，其对应关系在**Type-请求类型 对应表中**得以说明；“state”:value“‘name/value’对”表示服务器/客户端的运行状态，相同的值在客户端、服务器端有不同解释，其对应关系在**State-状态类型 对应表中**得以说明；“data”:value“‘name/value’对”表示具体传输内容，由于在JSON协议中值的情形很多，其详细设计将在下一部分中进行说明。

**示例：**

{“type”:100,”state”:0,

“data”:{ “UserName”:”王帆”,“NickName”:”Raven”,”UserID“:10000,”Psw”:”destiny25”}

}

这是一条客户端发向服务器的String，其中，“type”:100表示客户端向服务器发送了一条注册请求，“state”:0表示表示客户端正常，“data”:value表示注册时在数据库插入一行时写入数据库的信息。

1. **data的value组织**
2. **true/false**

一般用于服务器执行客户端请求之后的反馈信息，false表示出错，true表示成功。

**示例：**”data”:false

1. **JSONObject**

一般用于客户端向服务器发送数据库请求时，定义限制条件的“表项:项值“对的集合，或者服务器向客户端返回查询值时，定义结果的“表项:项值“对的集合。

**示例：**”data”:{“UserName”:”王帆”,“NickName”:”Raven”,”UserID“:10000,”Psw”:”destiny25”}

1. **JSONArray**

当存在返回多个value的需要时，采用JSONArray储存这些value的序列，其中一个典型的情形是服务器向客户端返回整张表。

**示例：**”data”:[{“UserName”:”王帆”,“NickName”:”Raven”,”UserID“:10000,”Psw”:”destiny25”},{“UserName”:”李元亨”,“NickName”:”MaydayorG”,”UserID“:10001,”Psw”:”meimima”}]

**type-请求类型 对应表**

|  |  |  |  |
| --- | --- | --- | --- |
| Type值 | 请求类型 | Type值 | 请求类型 |
| 100 | 注册 | **196** | 返回可接订单 |
| 101 | 登录 |  |  |
| 102 | 查询个人信息 |  |  |
| 103 | 提交大师兄意向 |  |  |
| 104 | 提交带哥意向 |  |  |
| 105 | 查询所有商品信息 |  |  |

**state-状态类型 对应表**

|  |  |
| --- | --- |
| State值 | 状态类型 |
| 0 | 正常 |
|  |  |
|  |  |
|  |  |
|  |  |