**iEat控制台概要设计**

GUI



网络结构



通信协议

/\* ===========================

\* 与Server通信的JSON-PDU格式

\* =========================== \*/

1.数据传输格式

Console和Server信息全部使用HTTPPOST传递，数据包使用JSON格式序列化，通过HTTP Request Payload提交请求，通过HTTP Response获取返回信息。

Sample PDU:

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

Request URL:http://localhost:8888/capi

Request Method:POST

Request Headers:

Connection:keep-alive

User-Agent:Http-client

Request Payload:

["connect","user1","password"]

Resonse PDU:

["000001" , "10001" , "00001" , [1,1],[12,1]]]

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

备注：其中Connect Request需要维护一个长期的HTTP连接(Connection:keep-alive)，服务器通过该通道传递点菜或者其他消息到console，该连接在空闲时要定期（没5分钟）发送心跳包，确保连接正常，一旦发现连接不正常，立刻发出警告。

数据请求目前包括以下调用：

2.Connect: 建立连接调用

Request Payload:

["connect","USER\_NAME#user1","TOKEN#password"]

字段描述：

1.Action ： connect

2.USER\_NAME : 用户名

3.TOKEN: 加密后的密码信息

Resonse PDU:

["LENGTH(135)","MD5\_HASH","000001" , "10001" , "00001" , [1,1],[12,1]]]

字段描述：

1.整个消息长度，包括这个字段本身，转化成ascii表示。

2.整个消息MD5 Hash值，用于确保消息发送正确，如果接收到MD5不正确的消息，需要丢弃，并发送重发请求，内容使用BASE64编码。

3.消息编号

4.服务员编号

5.桌编号

6.点菜信息

6.1 所点的菜编号

6.2 点菜数量

注：每次收到一个完整的Response PDU，就需要驱动打印机打印出一份点菜记录。

3.Resebd: 重发请求

Request Payload:

["resend","USER\_NAME#user1","TOKEN#password"]

字段描述：

1.Action ： resend

2.USER\_NAME : 用户名

3.TOKEN: 加密后的密码信息

Resonse PDU:

["OK|FAIL","MSG"]

字段描述：

1.成功或者失败

2.附加信息

4.SetState: 缺菜调用（PENDDING）