聊天室启动文件：

E:\web\51yj\include\lib\workerman-chat-for-win\start\_for\_win.bat

服务器和客户的交互文件：

E:\web\51yj\include\lib\workerman-chat-for-win\Applications\Chat\Events.php

规则：

聊天室运行环境简称 聊天内存。

进入聊天室时 需要携带一个聊天室的名字，如chat ，以下的所有互动都将默认会员在同一个聊天室内互动，简称chat.

聊天内存为每1个进入chat的人分配一个client\_id(长度20位)，存储于一个以自己client\_id命名的数组，以下简称uclient包，默认数据： ‘Fx0000000000001’($client\_id ) => [room\_id: ‘chat’]。每次重新进入chat，client\_id会变(递增生成)；

为了更好的跟踪会员，在会员每次进入chat时 必须带上自己的uid，聊天内存以session的形式存数于uclient包，如 $client\_id => [My\_uid: 123,room\_id: ‘chat’]

会员发起聊天时，必须带上对方的uid(传参名to\_uid)，然后从聊天内存根据其uid取出对方最新的client\_id（因为其随时可能重新登录,当对象离线时其会丢失to\_uid,只能通过to\_client反通知自己）， 聊天内存以session的形式存数于uclient包，储参名分别为：

talking\_to\_uid、talking\_to\_u\_clientid。

uclient包格式：

$client\_id => [My\_uid: 123,

talking\_to\_uid: 6,

talking\_to\_u\_clientid：’Fxxx0xxxxxxx0x0xx0x2’，

room\_id: ‘chat’]

流程举例:

会员a ,uid 6 ； 会员b,uid: 9； 为方便描述,clientid此省写长度统一以fff... 开头;

a上线时，（如打开聊天对象列表） 聊天内存生成 $client\_id = ‘fff...1’。

其uclient包为

‘fff...1’ => [My\_uid: 6,

talking\_to\_uid: 0,

talking\_to\_u\_clientid：’’，

room\_id: ‘chat’]

a与b聊天时，由于b未上线, a提交的所有内容都保存于服务器数据库。但每次a提交内容发送到服务器内存，都会更新a的uclient包，格式如下：

‘fff...1’ => [My\_uid: 6,

talking\_to\_uid: 9,

talking\_to\_u\_clientid：’’，

room\_id: ‘chat’]

b上线时, 聊天内存生成 $client\_id = ‘fff...2’, 通知所有与b建立了聊天窗口的会员，否则对方发送的clientid一直是0 ,b也无法接受a的即时信息。 实际上所有会员包括a上线时都要做一次身份通知。

身份通知内容：

1. 如果我是他的聊天对象，告诉他我的新client\_id
2. 如果我的当前聊天对象也是他 则给我的本地加他的clientid
3. 获取对方是否在线

// 获取房间内所有session

$chatSessionData = Gateway::getClientSessionsByGroup($room\_id);

foreach($chatSessionData as $temp\_clientid => $item\_) {

$tmp\_uid = isset($item\_['my\_uid']) ? $item\_['my\_uid'] : 0;

$talking\_to\_uid = isset($item\_['talking\_to\_uid']) ? $item\_['talking\_to\_uid'] : 0;

//如果我在对方的聊天列表中，告诉他我的新client\_id

if($my\_uid == $talking\_to\_uid) {

$new\_message\_login = array('type'=>'push\_my\_clientid\_to\_who\_talking\_to\_me', 'client\_id'=>$client\_id);

Gateway::sendToClient($temp\_clientid, json\_encode($new\_message\_login));

//如果我的当前聊天对象也是他 则给本地加他的clientid

if($message\_data['to\_uid'] == $tmp\_uid) {

$new\_message\_login = array('type'=>'push\_clientid\_to\_me', 'client\_id'=>$temp\_clientid);//更新我的clientid给他

Gateway::sendToCurrentClient(json\_encode($new\_message\_login));

}

}

//判断我的聊天对象 是否在线

if($tmp\_uid == $to\_uid) {

$new\_message\_login = array('type'=>'his\_online', 'his\_uid'=>$to\_uid);

Gateway::sendToCurrentClient(json\_encode($new\_message\_login));

$his\_client\_id = $temp\_clientid;//获取到他的client\_id

}

}

a对b发起聊天内容时，会提交本地保存的b的clientid,但是可能b的client可能已经更新(b重新登录过)。所以：a要判断b的真实身份。如果b的身份已经更新，要将新的clientid返回给a, 并且更新a的uclient包。

**表说明**

会员之间直接聊天

会员a可以与多人聊天，存储于

Crm\_webchat\_user\_obj 表

该表存储聊天各个对象和最后聊天时间，每2个人只生成一个记录，索引就叫做会话ID。

会员都有自己的会话ID表，存储于

Crm\_webcha\_user 表

该表存储会员最后的会话id,每次检测聊天内容，需要查询此表，提交uid查询出会员的聊天会话，首次查询时，传值最后聊天时间为null,如果获取到聊天记录后则取出最后聊天时间，并用于下次查询的对比，以判断是否有新的聊天内容。

聊天内容记录于

Crm\_webchat\_content

存储聊天内容表

会员未读信息只算未读会话数量

crm\_webchat\_user\_msg\_count