Openfire

# 1、架构



特点：

1. 模块化，模块可独立加载。
2. 命名清晰，和标准名称对应；
3. 所有业务通过配置来管理，配置数据启动的时候载入内存，运行时直接读取。
4. 事件监听通过AddListener来绑定。

# 2、Multi User Chat

## 2.1信息

XEP-0045: Multi-User Chat

涉及代码：

org.jivesoftware.openfire.muc

org.jivesoftware.openfire.muc.cluster

org.jivesoftware.openfire.muc.spi （service process implement？）

## 2.2数据库表格

### 2.2.1 ofmucservice

业务记录

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | Index | 备注 |
| serviceID | bigint(20) | key |  |
| subdomain | varchar(255) | primary key |  |
| description | varchar(255) |  |  |
| isHidden | tinyint(4) |  |  |

新增配置需系统重启后才能生效。

### 2.2.2 ofmucserviceprop

房间属性

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | Index | 备注 |
| serviceID | bigint(20) | primary key |  |
| name | varchar(100) | primary key | prop name |
| propValue | text |  | prop value |

### 2.2.3 ofmucroom

房间记录：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | Index | 备注 |
| serviceID | bigint(20) | primary key |  |
| roomID | bigint(20) | key | persistent / temporary(-1) don’t storage in DB. |
| creationDate | char(15) |  |  |
| modificationDate | char(15) |  |  |
| name | varchar(50) | primary key |  |
| naturalName | varchar(255) |  |  |
| description | varchar(255) |  |  |
| lockedDate | char(15) |  |  |
| emptyDate | char(15) |  |  |
| canChangeSubject | tinyint(4) |  |  |
| maxUsers | int(11) |  |  |
| publicRoom | tinyint(4) |  | public vs hidden |
| moderated | tinyint(4) |  | moderated vs unmoderated  需仲裁vs非仲裁 |
| membersOnly | tinyint(4) |  | members-only vs. open |
| canInvite | tinyint(4) |  |  |
| roomPassword | varchar(50) |  | password protected vs. unsecured |
| canDiscoverJID | tinyint(4) |  | non-anonymous, anonymous semi-anonymous |
| logEnabled | tinyint(4) |  |  |
| subject | varchar(100) |  |  |
| rolesToBroadcast | tinyint(4) |  |  |
| useReservedNick | tinyint(4) |  |  |
| canChangeNick | tinyint(4) |  |  |
| canRegister | tinyint(4) |  |  |

Room type：

1. Fully-Anonymous Room

用户JID不能被其他用户获得，包括Admins和owners  
Semi-Anonymous Room

用户JID只能被admins获取

Non-Anonymous Room

用户JID能被房间其他用户获取。

1. Hidden Room

用户不能通过search或service discover来查找

Public Room

房间能被所有用户搜索到。

1. Members-Only Room

只有在成员名单上的用户才能进入房间

Open Room

任何人都可进入房间

1. Moderated Room

只允许特殊人在房间里发言

Unmoderated Room

成员都可在房间里发送消息

1. Password-Protected Room

进入房间需输入正确的密码

Unsecured Room

进入房间不需设置密码

1. Persistent Room

当最后一个成员离开时房间不会被关闭

Temporary Room

当最后一个成员离开时房间会关闭

### 2.2.4 ofmucroomprop

房间属性

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | Index | 备注 |
| roomID | bigint(20) | primary key |  |
| name | varchar(100) | primary key | prop name |
| propValue | text |  | prop value |

### 2.2.5 ofmucaffiliation

记录会员身份：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | Index | 备注 |
| roomID | bigint(20) | primary key | room ID |
| jid | text | primary key | user JID |
| affiliation | tinyint(4) |  | owner(10)  admin(20)  member(30)  outcast(40)黑名单  none(50) |

### 2.2.6 ofmucmember

会员记录：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | Index | 备注 |
| roomID | bigint(20) | primary key |  |
| jid | text | primary key |  |
| nickname | varchar(255) |  |  |
| firstName | varchar(100) |  |  |
| lastName | varchar(100) |  |  |
| url | varchar(100) |  |  |
| email | varchar(100) |  |  |
| faqentry | varchar(100) |  |  |

### 2.2.7 ofmucconversationlog

记录群聊消息

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | Index | 备注 |
| roomId | bigint(20) |  |  |
| sender | text |  |  |
| nickname | varchar(255) |  |  |
| logTime | char(15) | primary key |  |
| subject | varchar(255) |  |  |
| body | text |  |  |

群聊消息先缓存在内存中，定时器驱动每隔5分钟存入数据库。

## 2.3 Affiliation和Role

Affiliation身份长期存在的身份，role角色是临时的，并在访问时改变。

四种Affiliation（身份）：

Owner（所有者）

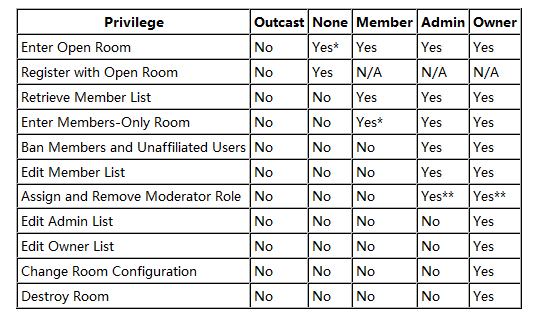
Admin（管理者）

Member（成员）：对会员制房间来说在白名单的用户，或注册到开放式房间的用户

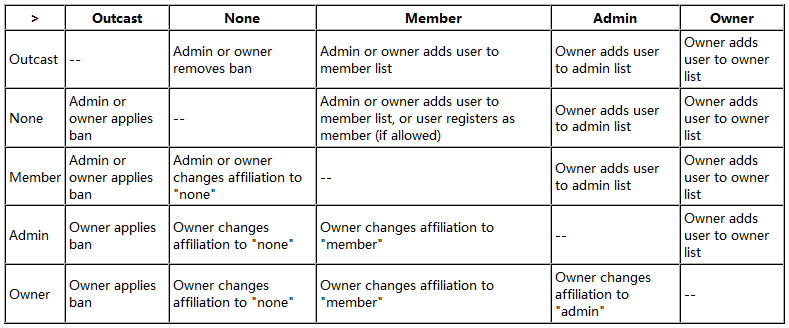
Outcast（黑名单）从房间里踢出并不能再进入room的用户。

None（缺席）

身份所对应的权限 Owner > Admin > Member > None > Outcast

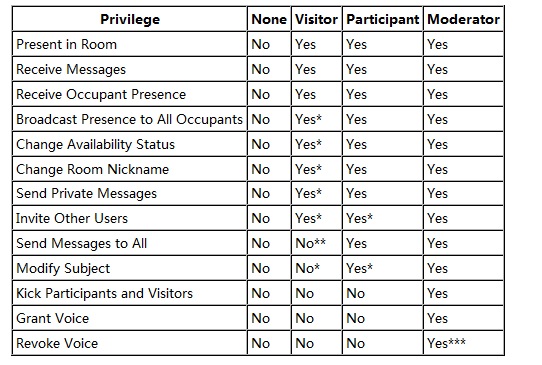


Affiliation的改变通常基于用户的行为，如注册为房间会员等。Affiliation的状态图：



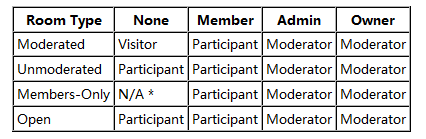
MCURole保存在内存里，*moderator仲裁者， participant参与者，visitor访问者, none黑名单*

*所拥有的权限 moderator > participant > visitor > none:*

**

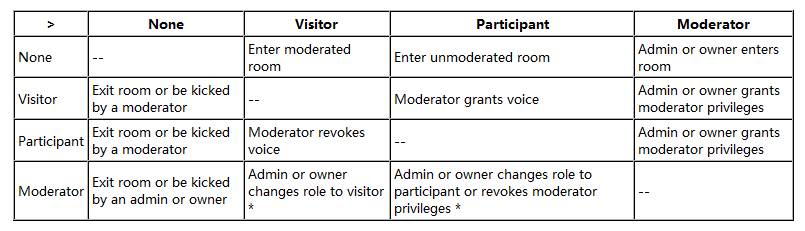
\*缺省值，配置可以更改该权限  
\*\*\*收回发言权，只适用于参与者，不适用于admin和owner

根据Affiliation获取最初的role。



事件可以更改occupant的Role，比如事件发生（加入房间或离开房间），Moderator更改Member的role。

Role状态图：



## 2.4 获取群的历史记录

1）by the maximum limit of characters to receive（完整的消息且字数小于客户端指定的maxchars）

2）limit by a maximum number of stanzas to receive

3）limit to receive only the messages before a given date

4）of the last X seconds

如何从两人聊天转为多人聊天：

1. 创建一个多人聊天室
2. 发送两人聊天记录到聊天室
3. 发送邀请给第二人、第三人

加入聊天室后的消息序列：



1、群聊消息存储在cache里，每隔5分钟定时由logging process存入数据库;

2、如成员没有发送任何消息给群，从群里清除IDLE成员（可配）(logConversationTask);（不需要清除，除非用户指定离开，才会离开群，否则发送push将消息推送到离线客户端）

3、30天后群里没有任何成员，从memory清除，但保存在数据库中(cleanupTask);

在一个cluster node上创建room, RemoteMCU处理和localMCU处理不同

需考虑：

5分钟没有存储相应数据到数据库，是否定时器问题？

room是否要增加Location信息（经度，纬度，地名等）

如何获取指定地点的room信息（是否要扩展XMPP参数？）

如何指定名称（room name，subject）搜索room

内存缓存的配置是针对一个用户还是所有用户？

Nickname冲突

如果成员达到上线值，可以把idle user踢走

业务的消息处理流程图。涉及的消息：presence、message

High Level Design （重要逻辑）

群聊的特色：

1. 支持所有的媒体（文本、图片、声音）发送
2. 加入所在位置周围的群聊
3. 以主题或位置搜索已有的聊天组，或者创建一个聊天组
4. 离线时不会退出聊天组，离线消息经过push通道通知客户端
5. 访客邀请加入，成员主动申请加入, 访客如何获得发言资格。
6. Public / non anonymous /un moderator/no password/ persistence
7. 什么情况下destory room？最后一个成员退出？destory后的历史聊天数据不再保存。

创建房间： 指定名字 ，主题，是否需要密码，

聊天

离开房间

根据经纬度查询房间：

<iq from='hag66@shakespeare.lit/pda'  
    id='zb8q41f4'  
    to='chat.shakespeare.lit'  
    type='get'>

<latitude> <longitude>  
  <query xmlns='http://jabber.org/protocol/disco#items'/>

</iq>