DiDi\_Plate Server端接口

## 1 项目介绍

Server端分为：Infobus（目前主要用于用户信息管理，包括用户注册，用户信息修改等等）和Openfire。 聊天功能基于xmpp的openfire开源项目。Openfire目前仅支持文字聊天功能，语音聊天需要后续详细研究。Openfire本身也有一些用户注册登录的功能，但是用在didi\_plate项目上不是很理想，也不方便以后进一步扩展管理didi\_plate这个项目。所以目前的是将Infobus里面的用户集成到openfire中。

项目Local Git url:

Infobus: http://cnpdgtdg211wx50:8080/GMGit/git/GMProjects/DIDIPlate\_Server.git

Openfire: <http://cnpdgtdg211wx50:8080/GMGit/git/GMProjects/GM_Openfire.git>

相关文档在Infobus工程/resource/document/目录底下

Git 需要用户名密码，目前有用户：

eric/12345

liumeng/12345

登录<http://cnpdgtdg211wx50:8080/GMGit/> 可查看所有git server上的项目

## 2 DiDi\_Plate 接口

接口按照用户从注册，登录我们的app, 然后开始聊天，到退出系统的整个流程。 具体顺序如下：

### 用户注册

客户端通过httpclient 发送post 请求到 infobus:

http://server:port/infobus/user/register.do, 请求参数如下：

|  |  |
| --- | --- |
| userName | spring04 //用户账户名 |
| name | 郭靖 |
| email | [test@gm.com](mailto:test@gm.com) |
| password | password //注册密码 |
| rPassword | password //确认密码 |
| userDetail.nickName | 酒肉穿肠 //用户昵称 |
| userDetail.phone | 18200000000//手机 |
| userDetail.plate | 沪ABCDE //车牌 |
| userDetail.sex | 1 //性别 |
| userDetail.signature | Tomorrow become better! |
| 用户签名 |  |
| userDetail.age | 28 //用户年龄 |
| userDetail.profession | 设计师 //用户年龄 |
| userDetail.location | 国定路四平路 //位置 |
| userDetail.idCard | //用户身份证号码 |

注册成功返回：

{"result":"0","data":{"userName":"spring04","createDate":null,"email":"test@gm.com","name":null,"status":0,"updateDate":null,"userDetail":{"userName":"spring04","birth":null,"idCard":null,"age":null,"location":null,"phone":"18200000000","nickName":"酒肉穿肠","photo":null,"plate":"沪ABCDE","profession":null,"sex":"1","signature":"Tomorrow become better!","email":null,"curVoiceUrl":null}},"msg":"注册成功!"}

注册失败会返回相应的错误消息：

{"result":"-2","data":[{"codes":["用户名已经存在..user.userName","用户名已经存在..userName","用户名已经存在..java.lang.String","用户名已经存在."],"arguments":null,"defaultMessage":null,"objectName":"user","field":"userName","rejectedValue":"spring04","bindingFailure":false,"code":"用户名已经存在."}],"msg":"Validation failed!"}

### 用户登录

用户登录openfire：

ConnectionConfiguration config = new ConnectionConfiguration(SERVER IP, PORT);

config.setSASLAuthenticationEnabled(false);

connection = new XMPPConnection(config);

connection.connect();

connection.login(username, password)

用户登录infobus并返回个人详细信息

Request url: <http://server:port/infobus/user/login>.do

Request params : username=spring01&password=password

### 修改个人状态（在线，离线等）

Presence presence = new Presence(Presence.Type.available);

presence.setStatus(status);

connection.sendPacket(presence);

### 用户图像修改

图像上传地址：

http://<Server IP : Port>/infobus/user/picUpload.do

上传参数：userName：spring01

上传文件域：imgFile

上传完返回：

{"result":"0","data":{"userName":"spring03","birth":null,"idCard":null,"age":null,"location":null,"phone":null,"nickName":null,"photo":"http://172.20.10.8/GMContent/userProfile/spring03\_1386143439724.png","plate":null,"profession":null,"sex":null,"signature":null,"email":null,"curVoiceUrl":null},"msg":"upload user photo successfully."}

错误也会返回相应的消息提示

### 用户签名修改

请求url：http://<Server IP : Port>/infobus/user/userDetailModify.do

参数：userDetail.userName, userDetail.signature

修改成功或失败返回相应json格式的提示信息

### 获得好友列表

public void getFriends(){

Roster roster = connection.getRoster();

Collection<RosterGroup> entriesGroup = roster.getGroups();

System.out.println("size--->"+entriesGroup.size());

for(RosterGroup group: entriesGroup){

Collection<RosterEntry> entries = group.getEntries();

System.out.println("group-->"+group.getName());

for (RosterEntry entry : entries) {

System.out.println("name-->"+entry.getName());

}

}

}

### 获得好友个人详细信息

Request url: <http://server:port/infobus/user/getUsersByUserNames>.do

Request params : username=spring01&username=spring02

Json response:

{"result":"0","data":{"spring02":{"userName":"spring02","birth":null,"idCard":null,"age":null,"location":null,"phone":"13800138000","nickName":"spring02","photo":"http://172.20.10.8/GMContent/userProfile/spring021386133809744.png","plate":"沪MI1231","profession":null,"sex":"1","signature":"心情不好","email":"test@gm.com","curVoiceUrl":null},"spring01":{"userName":"spring01","birth":null,"idCard":null,"age":null,"location":null,"phone":"13800138000","nickName":"spring01","photo":"http://172.20.10.8/GMContent/userProfile/spring011386133778833.png","plate":"沪MI1234","profession":null,"sex":"0","signature":"shit","email":"test@gm.com","curVoiceUrl":null}},"msg":"Success."}

### 通过车牌获得用户详细信息

Request url:

<http://server:port/infobus/user/getUserByPlate.do?>

Request param: plate=沪MI1234

Json response:

{"result":"0","data":{"userName":"spring01","createDate":null,"email":"test@gm.com","name":null,"status":0,"updateDate":null,"userDetail":{"userName":"spring01","birth":"1987-01-01","idCard":"1234567890111212","age":"28","location":"金皖路5号6","phone":"13800138000","nickName":"spring01","photo":"http://172.20.10.8/GMContent/userProfile/spring011386133778833.png","plate":"沪MI1234","profession":"平面设计","sex":"0","signature":"Happy every day!","email":null,"curVoiceUrl":null}},"msg":"找到注册的用户信息."}

找不到用户返回：

{"result":"-1","data":null,"msg":"车牌没有注册，找不到对应的用户."}

### 添加好友

Roster roster=conn.getRoster();

Collection<RosterEntry> it = roster.getEntries();

friends = new ArrayList<String>();

for(RosterEntry rosterEnter:it){

friends.add(rosterEnter.getUser());

}

### 给好友发送文字消息

ChatManager cm = XmppConnection.getConnection().getChatManager();

Chat mychat = cm.createChat(friendJid, null);

mychat.sendMessage("hello");

### 给好友发送语音消息

1. 先将语音上传给server, server返回一个语音url  
   上传request url: http://server:port/infobus/user/chatVoiceUpload.do

Request param: userName(发送消息者)

上传域: voiceFile

上传成功返回：

{"result":"0","data":{"userName":"spring01","birth":null,"idCard":null,"age":null,"location":null,"phone":null,"nickName":null,"photo":null,"plate":null,"profession":null,"sex":null,"signature":null,"email":null,"curVoiceUrl":"http://172.20.10.8/GMContent/voiceMsg/spring01\_1386208556460.xlsx"},"msg":"upload voice message successfully."}

1. 将类似”GM.VOICE.CHAT”的前缀+上述语音url拼接成一个字符串以文字的形式发送给要发送的用户，然后客户端根据消息前缀来解析该消息。

### 接收好友消息

conn.getChatManager().addChatListener(new ChatManagerListener() {

@Override

public void chatCreated(Chat chat, boolean createdLocally) {

chat.addMessageListener(new MessageListener() {

@Override

public void processMessage(Chat chat, Message message) {

String messageBody = message.getBody();

System.out.println("接收到信息："+messageBody);

}

});

}

})

### 退出登录

conn.disconnect();

### 其他接口

如搜索用户，模糊搜索等根据需要整理