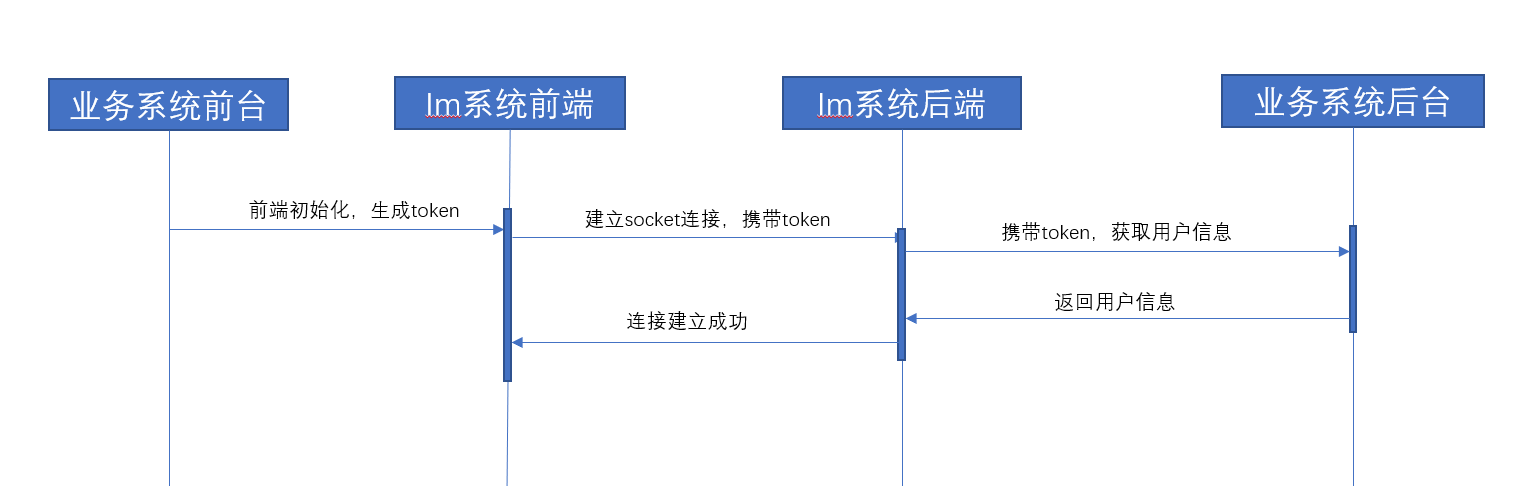
龙讯即时通讯接入说明文档

[**用户鉴权配置**](http://doc.5397078.cn/doc/java/index.html#/component/im/index?id=%e7%94%a8%e6%88%b7%e9%89%b4%e6%9d%83%e9%85%8d%e7%bd%ae)

为保证im的安全，以及用户信息的准确性， im需要与具体业务系统进行单点登录校验，需要您做一下工作：

1. 业务系统生成token，token在im前端配置
2. 业务系统提供根据token获取当前用户信息的接口
3. 在im系统中实现鉴权接口（鉴权时调用系统接口进行用户身份验证）



## WEB端配置

下面接口使用示例均为js api，安卓 与 IOS 接口地址与数据格式一致，具体API使用方法请参考 [**socket-io**](https://github.com/socketio/socket.io)

// 建立连接，参数说明

// 1. Ip端口 请与其他接口保持一致，建议使用常量

// 2. \_\_WEB\_IM\_TOKEN 会在登录接口返回，一般为token或ls-auth-token

// 3. sysType 所依托的服务，移动端使用oa\_mobile，PC端使用oa

var socketServer = "http://60.166.52.104:28080/?userUniqueKey=" + \_\_WEB\_IM\_TOKEN + "&sysType=oa\_mobile"

socket = io.connect(socketServer);

socket.on('connect', function() {

});

socket.on('disconnect', function() {

});

// 连接建立成功后请调用（user/info）拉取个人信息

**用户信息**

## 拉取个人信息（user/info）

socket建立成功，前端准备工作完成后由前端主动拉取用户个人信息（头像为空时请使用汉字的最后一个字）。

// 因为是拉取数据，不需要传递参数，提交空字符串即可

// friend 作为保留字段，目前尚未使用

// history 最近联系人，只保留20个

socket.emit('user/info', "", res => {

//res数据格式

{

"mine": {

"sign": "",

"organId": 100001496,

"id": 100001498,

"avatar": "",

"userId": 100001498,

"username": "张三",

"status": "online"

},

"friend": [

{

"id": 100005945,

"username": "潘梓薇",

"avatar": "",

"online": false

}

],

"history": [

{

"id": 100005945,

"username": "潘梓薇",

"avatar": "",

"online": false

}

],

"fileDownloadIp": "http://192.168.1.210:9000/" // 此处将文件预览地址返回，用户图片等附件的展示与下载

}

})

## 上线通知(user/online）

在用户进行第一次连接成功时会将平台内的所有在线用户通过这个接口推送过来，之后每次推送增量数据

// 前端监听上线信息

socket.on('user/online', res=>{

// res 数据格式, json的key是用户id, value 是上下线状态： 1上线， 0下线

{"100001498":1,"100007214":0}

})

**消息收发**

## 单聊消息监听 （p2p/msg）

socket.on('p2p/msg', data => {

//data数据格式：

{

"id": 0, //消息id

"sender": 0, // 消息发送人id

"senderName": "", // 消息发送人姓名

"receiver": 0, // 消息接收人id, 此处等于当前用户id

"content": "", // 消息内容

"sendDate": "2021-02-02 21:39:36", // 消息发送日期

"type": "", // 消息类型 （0：个人消息 1：群组消息）

"avatar": "", // 消息发送人头像，因为头像现在默认为姓名的最后一个字，所以此处为空

"groupName": "" // 群组名称

}

})

## 单聊消息发送 （p2p/msg）

let data = {

"receiver": 0, // 消息接收人id, 此处等于当前用户id

"receiverName": "", // 接收人姓名

"content": "" // 消息内容

};

socket.emit('p2p/msg', data, res => {

//res数据格式 {status:true, data: 时间戳}

//res数据格式 {status:true, message: "用户不在线"}

//res数据格式 {status:false, message: "登录状态异常/系统异常"}

})

## 设置消息已读 （p2p/read）

消息的已读针对用户进行，当打开用户的聊天窗口时需调用此接口，将用户消息设置为已读状态

// 对某个用户设置已读

let sender = 123123;

socket.emit('p2p/read', sender, res => {

//res数据格式 {status:true}

//res数据格式 {status:false, message: "登录状态异常/系统异常"}

})

## 最近聊天记录（p2p/newest）

应用场景：打开聊天窗口时显示的聊天记录可通过此接口获取

此接口只能获取最新聊天信息，如果需要更多历史聊天记录，需要调用历史聊天记录接口

// 获取聊天记录， 发送人或接收人中有一个id必须是自己，否则属于越权, 自己的id可以不传

let queryData = {

"receiver": 100001498, // 接收人

"sender": 100007214, // 发送人

"unidirectional": false, // 是否严格按照发送接收人获取数据，设置true时只能看到发送，不能看到回复， 缺省为false

"num": 4 // 获取条数， 缺省为15

};

socket.emit('p2p/newest', sender, res => {

//res数据格式

{

status:true,

data: [{

"id": 0,

"sender": 0,

"senderName": "",

"receiver": 0,

"receiverName": "",

"content": "",

"sendDate": "2021-02-02 21:39:36",

"type": "",

"avatar": "",

"groupName": ""

}]

}

//失败数据格式 {status:false, message: "登录状态异常/系统异常/您无权查看其他人的聊天记录！"}

})

## 历史聊天记录（p2p/history）

此接口可根据页码进行历史记录查询

// 获取聊天记录， 发送人或接收人中有一个id必须是自己，否则属于越权, 自己的id可以不传

let queryData = {

"receiver": 100001498, // 接收人

"sender": 100007214, // 发送人

"unidirectional": false, // 是否严格按照发送接收人获取数据，设置true时只能看到发送，不能看到回复， 缺省为false

"pageSize": 10, // 获取条数， 缺省为15

"pageIndex": 0 // 页码， 从 0 开始

};

socket.emit('p2p/history', sender, res => {

//res数据格式

{

"status": true,

"data": {

"data": [

{

"id": 4121,

"sender": 100007214,

"receiver": 100001498,

"content": "你好 龙讯",

"sendDate": "2021-02-02 14:01:40",

"type": "friend",

"avatar": ""

}

],

"total": 1006,

"pageSize": 4,

"pageIndex": 0,

"pageCount": 252,

"startNumber": 0

}

}

//res数据格式 {status:false, message: "登录状态异常/系统异常/您无权查看其他人的聊天记录！"}

})