

接口设计

接口均为原后台人员设计，如有疑问，可联系我进行沟通、修改。
由于接口均为反推，所以顺序并非开发顺序

编辑用户资料

- 地址 v1/accounts/{uuid}
- 请求方式：PUT
- 参数

名称	提交方式	数据格式	说明
uuid	path	str	
avatar	body	str	头像
nickname	body	str	昵称
birthday	body	str	生日
gender	body	int	1 2
region	body	str	所在区域，城市
height	body	str	身高
weight	body	str	体重
job	body	str	职业
intro	body	str	签名
figure	body	str	身材
emotion	body	str	情感状况
child	body	str	是否有孩子
education	body	str	学历
income	body	str	年收入
smoke	body	str	是否吸烟
drink	body	str	是否饮酒

- 请求方式与返回值
请求方式及返回值参考注册、认证的填写信息相关接口，如果某个参数没有传递，则维持服务器中保存的数据

删除照片

- 地址 v1/accounts/{uuid}/resource
- 请求方式：delete
- 参数

名称	提交方式	数据格式	说明
----	------	------	----

名称	提交方式	数据格式	说明
uuid	path	str	
uuids	body	map	照片的uuid集合

- 请求示例

```
//delete v1/accounts/{uuid}/resource
{
  "uuids":[
    123,
    456,
    789
  ]//照片的uuid集合
}
```

获取本人信息

- 地址 v1/accounts/{uuid}
- 请求方式: get
- 参数

名称	提交方式	数据格式	说明
uuid	path	str	

- 请求示例

```
//post v1/accounts/本人uid
```

- 返回值

```
{
  "uuid": "用户唯一标识",
  "nickname": "用户昵称",
  "gender": "1|2", //1男2女
  "avatar": "", //用户头像
  "member": {
    "status": "true|false" //是否购买了会员
  },
  "auth_user": {
    "status": "true|false" //是否认证过
  },
  "age": 18, //年龄
  "user_detail": {
    "intro": "个人签名",
    "region": "所在地区",
    "height": "身高",
    "weight": "体重",
    "reg_schedule": "100|101|102|103",
    //**** 这里需要特别注意, 这里联动了注册所在步骤,
    //主要是避免用户注册到一半杀掉app, 重新注册时返回至之前杀掉的页面
    //100: 完成注册
    //101: 未选择性别
    //102: 未填写头像、昵称、生日
    //103: 未选择想要的关系
    //第一次注册返回101
    //这里我们只用100, 101, 102即可, 103暂时删除
  },
}
```

```
"photo":{
  "type":"image|video",
  "resource":"资源地址",
  "width":"图片宽度",
  "height":"视频宽度",
  "preview":"图片预览图",
  "uuid":"资源唯一标识",
  "cover":"加了高斯模糊的图片",//这里应该是只有阅后即焚的资源才需要
  "pay_type":"red_packet|fire|free",//red_packet: 收费视频, fire: 阅后即焚图片, free: 免费
  "status":"100|101",//阅后即焚状态 100未读, 101已读 这里注意阅后即焚每天重置一次
  "small_cover":"加了告诉模糊的预览图",
},
"job":{
  "uuid":"",
  "name":"职业"
},
"netease":{
  "accid":"云信账号",
  "token":"云信密码"
},
"follow_count":"0",//我关注了多少人
"followed_count":"0",//多少人关注了我
"birthday":"1990-03-19",//生日
"wechat":{
  "number":"微信号|本项目为whatsapp号码",
  "status":"pass",//本项目中始终返回pass即可
  "lock":"true|false",//是否隐藏了号码
},
"is_register":"0|1",//是否是第一次注册的用户
"has_video":"true|false",//是否上传了视频 视频只有女生可以上传
"qa":{
  "url":"在线客服地址",//这里我们之前接了第三方客服系统, 是个web版本的im, 没有可以不用管
},
"account_state":"normal|destroying|destroyed",//始终返回normal即可, 我们暂时不做销毁功能
"UserInfoDetailExtra":{
  //以下字段都是一些用户选择完的标签
  "emotion":{"情感状态标签",
    "uuid":"唯一标识",
    "name":"名称"
  },
  "child":{"是否有孩子",
    "uuid":"唯一标识",
    "name":"名称"
  },
  "education":{"学历",
    "uuid":"唯一标识",
    "name":"名称"
  },
  "figure":{"是否有孩子",
    "uuid":"唯一标识",
    "name":"名称"
  },
  "income":{"收入",
    "uuid":"唯一标识",
    "name":"名称"
  },
  "smoke":{"是否吸烟",
    "uuid":"唯一标识",
    "name":"名称"
  },
  "drink":{"是否喝酒",
    "uuid":"唯一标识",
    "name":"名称"
  },
},
//兴趣爱好, 这是个多选标签
"hobbys":[
  {
    "uuid":"唯一标识",
    "name":"抽烟"
  },
  {
    "uuid":"唯一标识",
    "name":"喝酒"
  }
]
```

```
    },
    {
      "uuid": "唯一标识",
      "name": "烫头"
    }
  ],
  "invite_info": {
    "invite_qrcode_url": "邀请码二维码地址"
  },
  "photo_count": "0", // 用户上传的照片数量，不包含头像
}
```

获取注册、编辑资料时的一些下拉选项

- 地址 v1/accounts/{uuid}/detail_extra
- 请求方式: get
- 参数

名称	提交方式	数据格式	说明
uuid	path	str	

- 请求示例

```
//get v1/users/对方用户uid
```

- 返回值

```
{
  "emotion": [{"uuid": "唯一编号", "name": "选项名"}], // 情感状况
  "child": [{"uuid": "唯一编号", "name": "选项名"}], // 是否有孩子
  "education": [{"uuid": "唯一编号", "name": "选项名"}], // 学历
  "income": [{"uuid": "唯一编号", "name": "选项名"}], // 年收入
  "figure": [{"uuid": "唯一编号", "name": "选项名"}], // 身材
  "smoke": [{"uuid": "唯一编号", "name": "选项名"}], // 是否吸烟
  "drink": [{"uuid": "唯一编号", "name": "选项名"}], // 是否喝酒
  "hobby": [{"uuid": "唯一编号", "name": "选项名"}], // 爱好
  "job": [{"uuid": "唯一编号", "name": "选项名"}] // 职业
}
```

获取其他人信息

- 地址 v1/users/{uuid}
- 请求方式: get
- 参数

名称	提交方式	数据格式	说明
uuid	path	str	

- 请求示例

```
//get v1/users/对方用户uid
```

- 返回值
与上述获取个人信息一致，但是要注意部分字段需隐藏或处理，
如wechat字段中的number,再本人未解锁对方时应格式化为19，等特殊隐藏样式

是否可以查看对方

- 需求描述
男用户为非会员时，每天可以查看20次女性用户，此接口用来查询是否有权限查看。会员账户每天可查看100次。
- 地址 v1/users/{uuid}/is_look
- 请求方式：get
- 参数

名称	提交方式	数据格式	说明
uuid	path	str	

- 请求示例

```
//get  v1/users/{uuid}/is_look
```

- 返回值

```
{
  "code":1007,
  "message":"success",
  "data":{
    "is_looked":"true|false",//是否能查看对方
    "count":"0",//剩余次数
    "alert_status":"true|false"//是否弹出提醒
  }
}
```

消费查看用户次数

- 需求描述
每次查看一个用户消费一次，同一个人可以反复查看，每天重置次数。
- 地址 v1/users/{uuid}/look
- 请求方式：get
- 参数

名称	提交方式	数据格式	说明
uuid	path	str	

- 请求示例

```
//get  v1/users/{uuid}/look
```

- 返回值

```
{
  "code":1007,
  "message":"success",
}
```

获取评价的标签列表

- 需求描述
用来获取评价他人打分的列表
- 地址 v1/tags/user-evaluate/{uuid}
- 请求方式: get
- 参数

名称	提交方式	数据格式	说明
uuid	path	str	

- 请求示例

```
//get  v1/tags/user-evaluate/{uuid}
```

- 返回值

```
{
  "code":1007,
  "message":"success",
  "data":[
    {
      "uuid":"标签id",
      "name":"颜值",// 每个标签的名称，如身材、颜值等
    }
  ]
}
```

评价用户

- 需求描述
用以评价用户时提交评价分数,解锁女生后才能评价，并且只能评价一次。
- 地址 v1/users/{uuid}/tags?type=evaluate
- 请求方式: post
- 参数

名称	提交方式	数据格式	说明
uuid	path	str	对方uuid
uuid	body	str	标签的uuid
star	body	float	分数，最小单位0.5f

- 请求示例

```
//post v1/users/{uuid}/tags?type=evaluate
[
  {
    "uuid": "标签id",
    "star": "4.5"
  }
]
```

获取对方的评价信息

- 地址 v1/accounts/{uuid}/evaluate
- 请求方式: get
- 参数

名称	提交方式	数据格式	说明
uuid	path	str	

- 请求示例

```
//get v1/accounts/{uuid}/evaluate
```

- 返回值

```
{
  "code": 1007,
  "message": "success",
  "data": {
    "count": "10+", // 应该是有多少人评价过, N人以上显示+
    "data": [
      {
        "star": "0f", // 每个标签的分数
        "tag": "颜值" // 标签名称
      }
    ]
  }
}
```

举报用户

举报用户首先客户端会利用图片上传接口上传图片，之后将获取到的图片地址提交到report接口，上传图片的type为report

- 地址 v1/report/{uuid}
- 请求方式: POST
- 参数

名称	提交方式	数据格式	说明
uuid	path	str	被举报人的uuid
content	body	str	举报的具体内容
photos	body	list	上传的图片地址

- 请求示例

```
//post v1/report/{uuid}

{
  "content": "这人NT",
  "photos": ["1.png", "2.png"]
}
```

意见反馈

意见反馈首先客户端会利用图片上传接口上传图片，之后将获取到的图片地址提交到report接口，上传图片的type为feedback

- 地址 v1/feedback/{uuid}
- 请求方式：POST
- 参数

名称	提交方式	数据格式	说明
uuid	path	str	被举报人的uuid
content	body	str	举报的具体内容
photos	body	list	上传的图片地址

- 请求示例

```
//post v1/feedback/{uuid}

{
  "content": "这人NT",
  "photos": ["1.png", "2.png"]
}
```

关注用户

- 地址 v1/follow
- 请求方式：POST
- 参数

名称	提交方式	数据格式	说明
uuids	body	list	可多个同时关注

- 请求示例

```
//post v1/follow

{
  "uuids": [
    123123,
    213123,
    213123123,
  ]//用户的uuid
}
```


取消关注

- 地址 /v1/unfollow/users/{uuid}
- 请求方式: delete
- 参数

名称	提交方式	数据格式	说明
uuid	path	str	

- 请求示例

```
//delete /v1/unfollow/users/{uuid}
```

拉黑某人

- 地址 v1/blacklists
- 请求方式: post
- 参数

名称	提交方式	数据格式	说明
uuids	body	list	可多个拉黑

- 请求示例

```
//post v1/blacklists

{
  "uuids":[
    123123,
    213123,
    213123123,
  ]//用户的uuid
}
```

取消拉黑

- 地址 /v1/blacklists/users/{uuid}
- 请求方式: delete
- 参数

名称	提交方式	数据格式	说明
uuid	path	str	

- 请求示例

```
//delete /v1/blacklists/users/{uuid}
```