

- Author: Ji Yang
- Contact: yangji9181[AT]gmail.com
- Changelog:
 - 2014.07.10 Initial version.
 - 2014.08.01 Edited by Ji Yang

一,URL说明

url相对路径	HTTP variable	功能
/searchforuser/	GET	搜索用户
/sendfriendrequest/	POST	发送好友请求
/getfriendrequest/	POST	获取好友请求
/acceptfriendrequest/	POST	通过好友请求
/rejectfriendrequest/	POST	拒绝好友请求
/getfriendlist/	POST	获取好友列表
/isfriend/	GET	判断任意两个用户是否为好友
/deletefriend/	POST	删除好友

二，数据格式

(1)搜索用户： /searchforuser/ -X GET

请求数据格式：

```
{
  query:  string,      查询关键字, required
}
```

返回数据格式：

```
JSON:
{
  status:    string,
  message:   string,
  data: {
    id:      integer,
    email:   string,
    nick_name: string,
    first_name: string,
    last_name: string,
    gender:   string,
    phone:    string,
    uniserity: integer, //university id, 可用来检索具体学校信息
    major:    integer  //major id, 可用来检索具体专业信息
  }
}
```

(2)发送好友请求: /sendfriendrequest/ -X POST

请求数据格式:

```
{
  token: string,      //required
  receiverid: integer //required
}
```

返回数据格式:

```
JSON:
{
  status:    string,
  message:   string,
}
```

(3)获取好友请求: /getfriendrequest/ -X GET

请求数据格式:

```
{
  token: string, //登录时获得, required
}
```

返回数据格式:

```
JSON:
{
  status:    string,
  message:   string,
  data:      {
    id:       integer,    //request id
    created:  string,     //发送时间
    receiver: integer,
    sender:   integer
  }
}
```

(4)通过好友请求: /acceptfriendrequest/ -X POST

请求数据格式:

```
{
  token:      string    //登录时获得, required
  requestid:  integer    //通过getfriendrequest获得, required
}
```

返回数据格式:

```
JSON:
{
  status:    string,
  message:   string,
}
```

(5)拒绝好友请求: /rejectfriendrequest/ -X POST

请求数据格式:

```
{
  token:      string,    //登录时获得, required
  requestid:  integer    //通过getfriendrequest获得, required
}
```

返回数据格式:

```
JSON:
{
    status:    string,
    message:   string,
}
```

(6)获取好友列表: /getfriendlist/ -X GET

请求数据格式:

```
{
    token:      string,      //登录时获得, required
}
```

返回数据格式:

```
JSON:
{
    status:    string,
    message:   string,
    data:{
        id:     integer,      //relation id
        user1:  integer,
        user2:  integer
    }
}
```

(7)判断任意两个用户是否为好友: /isfriend/ -X GET

请求数据格式:

```
{
    user1: integer,    //required
    user2: integer    //required
}
```

返回数据格式:

```
JSON:
{
  true or false
}
```

(8)删除好友: /deletefriend/ -X POST

请求数据格式:

```
{
  token:      string,      //登录时获得, required
  friendid:   integer      //required
}
```

返回数据格式:

```
JSON:
{
  status:     string,
  message:    string
}
```

三, 返回数据说明

1, status 字段

status code	功能
"success"	请求成功
"fail"	请求失败 (客户端提交数据错误等)
"error"	服务器错误 (数据库不一致等异常情况)

2, message 字段

用于请求发生错误时返回错误信息

3, data

部分请求中用于返回客户端所需的数据。

四，解释说明

- 用户只能操作与自己相关的好友请求或好友列表的事件，否则会引发'permission denied'错误
- 所有required字段不传入会引发'XXX data format error'错误

1， /searchforuser/ 搜索用户

不需要传入token，只需传入一个query，query只能是一个词，可以是一个email，一个nick_name，一个first_name，一个last_name，一个gender或一个phonenummer。需要更灵活的搜索功能请与后端讨论。

2， /sendfriendrequest/ 发送好友请求

好友请求只能发送一次，被拒绝后可以再次发送，不能对好友发送好友请求，不能对自己发送好友请求

3， /getfriendrequest/ 获取好友请求

得到所有自己未拒绝且未通过的好友请求

4， /acceptfriendrequest/ 通过好友请求

通过好友请求并建立好友关系，只能通过发送给自己且尚未拒绝或通过的好友请求

5， /rejectfriendrequest/ 拒绝好友请求

拒绝好友请求，只能拒绝发送给自己且尚未拒绝或通过的好友请求

6， /getfriendlist/ 获取好友列表

获取自己的好友列表，返回数据中可能user1, user2分别对应自己和好友，也可能user2, user1分别对应自己和好友，前端显示时需要简单判断

7， /isfriend/ 判断任意两个用户是否为好友

不需要token，只需传入任意两个用户id，注意返回数据仅仅是一个true或者false

8， /deletefriend/ 删除好友

只能删除已经建立好友关系的好友