



API 接口文档

ThinkSNS V3R1

智士软件（北京）有限公司

ZhiShiSoft (Beijing) Co., Ltd.

修订记录

日期	修订版本	描述	作者
2012/12/29	0.1	建立文档	胡建龙

目录

1	项目信息	6
2	文档介绍	6
2.1	编写目的	6
3	接口设计	6
3.1	接口基本说明	7
3.2	接口调用	7
4	名词解释	8
5	接口定义	8
5.1	认证	9
5.1.1	【request_key】 Oauth/request_key	9
5.1.2	【用户认证】 Oauth/authorize	9
5.2	消息	10
5.2.1	【通知总数】 Notifytion/get_notify_count	10
5.2.2	【消息类型及数量】 Notifytion/get_notifyList	11
5.2.3	【设置消息已读】 Notifytion/set_notify_read	13
5.2.4	【设置对话已读】 Notifytion/set_message_read	14
5.2.5	【未读系统消息】 Notifytion/get_system_notify	14
5.3	系统	错误! 未定义书签。
5.3.1	【添加收藏】 System/favorite_create	错误! 未定义书签。

5.3.2	【取消收藏】 System/favorite_destroy	错误! 未定义书签。
5.3.3	【评论接口】 System/comment	错误! 未定义书签。
5.3.4	【分享接口】 System/share	错误! 未定义书签。
5.3.5	【获取评论】 Message/get_comments	错误! 未定义书签。
5.4	微博	16
5.4.1	【全部微博】 WeiboStatuses/public_timeline	16
5.4.2	【关注微博】 WeiboStatuses/friends_timeline	22
5.4.3	【发布微博】 WeiboStatuses/user_timeline	22
5.4.4	【@我的微博列表】 WeiboStatuses/mentions_feed	23
5.4.5	【微博详情】 WeiboStatuses/show	24
5.4.6	【发文字微博】 WeiboStatuses/create	24
5.4.7	【发图片微博】 WeiboStatuses/upload	27
5.4.8	【删除微博】 WeiboStatuses/destroy	28
5.4.9	【转发微博】 WeiboStatuses/repost	28
5.4.10	【评论列表】 WeiboStatuses/comments	29
5.4.11	【评论我的】 WeiboStatuses/comments_to_me	32
5.4.12	【我评论的】 WeiboStatuses/comments_by_me	39
5.4.13	【发出评论】 WeiboStatuses/comment	39
5.4.14	【收藏微博列表】 WeiboStatuses/favorite_feed	40
5.4.15	【添加收藏】 WeiboStatuses/favorite_create	43
5.4.16	【取消收藏】 WeiboStatuses/favorite_destroy	43

5.4.17	【收藏的话题列表】 WeiboStatuses/weibo_topics	44
5.4.18	【关注话题】 WeiboStatuses/weibo_topics_create	44
5.4.19	【取消关注】 WeiboStatuses/weibo_topics_destroy	45
5.4.20	【搜索微博】 WeiboStatuses/weibo_search_weibo.....	45
5.4.21	【搜索用户】 WeiboStatuses/weibo_search_user	46
5.5	用户	46
5.5.1	【用户信息】 User/show	46
5.5.2	【上传头像】 User/upload_face.....	52
5.5.3	【添加关注】 User/follow_create	52
5.5.4	【取消关注】 User/follow_destroy.....	53
5.5.5	【粉丝列表】 User/user_followers.....	54
5.5.6	【关注列表】 User/user_following.....	58
5.6	聊天	60
5.6.1	【对话列表】 Message/get_message_list.....	60
5.6.2	【对话详情】 Message/get_message_detail.....	62
5.6.3	【发布消息】 Message/create	63
5.6.4	【回复消息】 Message/reply	64
5.6.5	【删除消息】 Message/destroy	64
5.7	频道	65
5.7.1	【频道列表】 Channel/get_all_channel.....	65
5.7.2	【频道微博】 Channel/get_channel_feed.....	66

5.8	微吧	67
5.8.1	【微吧列表】 Weiba/get_all_weiba	67
5.8.2	【帖子列表】 Weiba/get_posts	67
5.9	签到	78
5.9.1	【签到信息】 Checkin/get_check_info	78
5.9.2	【签到操作】 Checkin/checkin	79

1 项目信息

项目名称	ThinSNSV3		
客户名称		项目经理	
供应商名称	智士软件（北京）有限公司	实施经理	

2 文档介绍

2.1编写目的

本文档是为 ThinSNSV3 产品 API 说明文档，为 ThinsnsV3 客户端开发者、系统继承者提供演示说明和 API 说明文档。

3 接口设计

3.1接口基本说明

API 核心文件在 ThinsnsV3/core/OpenThinsnsV3/Api.class.php , 其他应用 API 存放于 ThinsnsV3/addons/api 。 Api 文档存放于 SVN , 文件路径为 : ThinsnsV3/ThinsnsV3Client/手册文档。ThinsnsV3Api 以应用为划分模块, 命名为 : { MODULE_NAME } Api.class.api , { MODULE_NAME } 为应用名。

API 为 WeiboStatusesApi.class.php , 每个 API 必须继承于核心 API , 如 :

```
<?php
class WeiboStatusesApi extends Api{
    function public_timeline(){
    }
}
?>
```

3.2接口调用

根据 HTTP 协议, 通过 GET/POST 形式, 接口将返回 JSON 格式数据。接口调用过程如下 :

1) request_key token

- + 地址 : http://SITE_URL/index.php?app=api&mod=Oauth&act=request_key
- + 参数 : 无
- + 返回值 : JSON 格式的 request token

2) 获取用户的 oauth_token 和 oauth_token_secret、uid

- + 地址 : http://SITE_URL/index.php?app=api&mod=Oauth&act=authorize
- + 参数类型 : POST

+ 参数：

- uid：[必须] 经过 DES 加密的用户 ID 或用户 Email（使用第1步的 request token 做 DES 加密）

- passwd：[必须] 经过 DES 加密的用户密码（使用第1步的 request token 做 DES 加密）

+ 返回值：JSON 格式的 oauth_token、oauth_token_secret、uid 数据

3) 调用 Api

+ 地址：

http://SITE_URL/index.php?app=api&mod=MODULE_NAME&act=ACT_NAME&oauth_token=OAUTHER_TOKEN&oauth_token_secret=OAUTH_TOKEN_SECRET

+ 参数类型：参考 Api 定义

+ 参数：参考 API 定义

+ 返回值：参考 API 定义

4 名词解释

名词/术语	名词解释
时间戳：	时间戳是自 1970 年 1 月 1 日（00:00:00 GMT）以来的秒数。它也被称为 Unix 时间戳（Unix Timestamp）

5 接口定义

注：（返回结果结为 JSON,为方便阅读和传输）

5.1 认证

5.1.1 【request_key】Oauth/request_key

接口名称：	Oauth/request_key	
接口描述：	返回应用的 request_key 。	
演示地址：		
输入参数：	(string) format , 返回格式可选 json/php/test 格式 (默认 json 格式)	
输出结果：	<pre>array(1) { [0] => string(8) "thinksns" }</pre>	
字段说明	成功时：	返回用户加密的字串
	失败时：	错误信息
其他说明：	即用来加密的 key ,系统默认为 thinksns ,实际上应该根据不同的请求的应用 , 返回不同的信息开发者请手工修改这个值	

5.1.2 【用户认证】Oauth/authorize

接口名称：	Oauth/authorize	
接口描述：	通过 Oauth 进行认证	
演示地址：		
输入参数：	(string) uid , 帐号 DES 加密 (必填) (string) passwd , 密码双重加密 (必填) (string) format , 返回格式可选 json/php/test 格式 (默认 json 格式)	

输出结果：	<pre>array(3) { ['uid'] => string(5) "10086", ['oauth_token']=>string(32)" 1697d5e169f72520025943258b70959d", ['oauth_token_secret']=>string(32)" 3c377ce4c5ed340a77d2eb9f215cd511", } array(2){ ['message'] => string(4) "认证失败", ['code']=>string(5)" 00001", }</pre>	
字段说明	成功时：	uid：用户 UID oauth_token：用户 oauth_token oauth_token_secret：用户 oauth_token_secret
	失败时：	Message:错误信息 Code:错误代号
其他说明：	1,双重加密是先进行 MD5 加密 在进行 DES 加密。加密的 key 用 request_key。 2 ,oauth_token 和 oauth_token_secret 为请求 Api 必带参数	

5.2 消息

5.2.1 【通知总数】 Notifyon/get_notify_count

接口名称：	Notifyon/get_notify_count
接口描述：	获取通知总数
演示地址：	http://192.168.1.200/ThinsnsV3/index.php?app=api&mod=AppList&act=get_notify_count&oauth_token=1e52300bf6576b2efe63a7fa4ec37b9c&oauth_token_secret=cb730170da8c17dc7f5683669303a09f&format=json
输入参数：	(int) mid，当前登录用户 UID（选填，默认当前用户）

	(string) format , 返回格式可选 json/php/test 格式 (默认 json 格式)	
输出结果 :	Int 整数	
字段说明	成功时 :	大于零的整数
	失败时 :	0
其他说明 :		

5.2.2 【消息类型及数量】Notifyon/get_notifyList

接口名称 :	Notifyon/get_notifyList	
接口描述 :	获取当前用户消息提醒类型和总数	
演示地址 :	http://192.168.1.200/ThinsnsV3/index.php?app=api&mod=AppList&act=get_notifyList&oauth_token=1e52300bf6576b2efe63a7fa4ec37b9c&oauth_token_secret=cb730170da8c17dc7f5683669303a09f&format=json	
输入参数 :	(int) mid , 当前登录用户 UID (选填 , 默认当前用户) (string) format , 返回格式可选 json/php/test 格式 (默认 json 格式)	
输出结果 :	<pre>array(3) { [0] => array(5) { ["type"] => string(4) "atme" ["name"] => string(7) "@我的" ["icon"] => string(8) "AtMeIcon" ["count"] => int(0) ["data"] => string(0) "" } [1] => array(5) { ["type"] => string(7) "comment" ["name"] => string(6) "评论" ["icon"] => string(11) "CommentIcon" ["count"] => NULL } }</pre>	

```

    ["data"] => string(0) ""
}
[2] => array(5) {
    ["type"] => string(7) "message"

    ["name"] => string(6) "私信"

    ["icon"] => string(11) "MessageIcon"
    ["count"] => int(0)
    ["data"] => array(16) {
        ["list_id"] => string(1) "7"
        ["member_uid"] => string(5) "14983"
        ["new"] => string(1) "0"
        ["message_num"] => string(1) "1"
        ["ctime"] => string(1) "0"
        ["list_ctime"] => string(10) "1337411282"
        ["from_uid"] => string(5) "27842"
        ["type"] => string(1) "1"
        ["title"] => string(9) "123123123"
        ["member_num"] => string(1) "2"
        ["min_max"] => string(11) "14983_27842"
        ["mtime"] => string(10) "1337410679"
        ["last_message"] => string(57)
"a:2:{s:8:\"from_uid\";i:27842;s:7:\"content\";s:7:\"1111
111\";}"
        ["content"] => string(7) "1111111"

        ["from_urname"] => string(15) "数学课代表"

        ["form_face"] => string(63)
"http://localhost/work/st/data/upload/avatar/6d/96/c
4/middle.jpg"
    }
}
}

```

字段说明	成功时：	Type：通知类型 Name：通知名字 Icon：通知图标 Count：通知类型对应数量 Data：数据 (此处补充聊天的字段信息)
	失败时：	空数组
其他说明：		

5.2.3 【设置消息已读】Notifyon/set_notify_read

接口名称：	Notifyon/set_message_read
接口描述：	设置消息为已读
演示地址：	http://192.168.1.200/ThinsnsV3/index.php?app=api&mod=AppList&act=set_notify_read&oauth_token=1e52300bf6576b2efe63a7fa4ec37b9c&oauth_token_secret=cb730170da8c17dc7f5683669303a09f&format=json
输入参数：	(int) mid，当前登录用户 UID (string)data['type']，类型（必填，类型包括：atme、comment、new_follower） (string) format，返回格式可选 json/php/test 格式（默认 json 格式）
输出结果：	Int 类型整数

字段说明	成功时：	1
	失败时：	0
其他说明：		

5.2.4 【设置对话已读】Notifyon/set_message_read

接口名称：	Notifyon/set_message_read	
接口描述：	设置某对话为已读	
演示地址：	http://192.168.1.200/ThinsnsV3/index.php?app=api&mod=AppList&act=set_message_read&oauth_token=1e52300bf6576b2efe63a7fa4ec37b9c&oauth_token_secret=cb730170da8c17dc7f5683669303a09f&format=json	
输入参数：	(int) mid , 当前登录用户 UID (int)list_id , 当前对话 ID (必填) (string) format , 返回格式可选 json/php/test 格式 (默认 json 格式)	
输出结果：	Int 类型整数	
字段说明	成功时：	1
	失败时：	0
其他说明：		

5.2.5 【未读系统消息】Notifyon/get_system_notify

接口名称：	Notifyon/get_system_notify	
接口描述：	返回未读的系统消息	

演示地址：	<code>http://192.168.1.200/ThinsnsV3/index.php?app=api&mod=AppList&act=search_app&oauth_token=1e52300bf6576b2efe63a7fa4ec37b9c&oauth_token_secret=cb730170da8c17dc7f5683669303a09f&format=json</code>
输入参数：	<p>(int) mid , 当前登录用户 UID (选填 , 默认当前用户)</p> <p>(int) since_id , 起始应用 ID (选填)</p> <p>(int) max_id , 最大应用 ID (选填)</p> <p>(int) count , 分页显示时 , 指定每页显示条数 (选填 , 默认 20)</p> <p>(int) page , 分页显示时 , 指定获取的页码 (选填 , 默认取第 1 页)</p> <p>(string) format , 返回格式可选 json/php/test 格式 (默认 json 格式)</p>
输出结果：	<pre>array(1) { [0] => array(5) { ["type"] => string(6) "notify" ["name"] => string(12) "系统私信" ["icon"] => string(10) "NotifyIcon" ["count"] => string(1) "1" ["data"] => array(15) { ["list_id"] => string(2) "55" ["member_uid"] => string(5) "27852" ["new"] => string(1) "1" ["message_num"] => string(1) "1" ["ctime"] => string(1) "0" ["list_ctime"] => string(10) "1339080093" ["is_del"] => string(1) "0" ["from_uid"] => int(1) ["type"] => string(1) "3" ["title"] => string(57) "欢迎使用 ThinsnsV3 , 与同事高效协作 , 快乐沟通" ["member_num"] => string(1) "2" ["min_max"] => string(7) "1_27852" ["mtime"] => string(10) "1339080093" ["last_message"] => array(4) { ["from_uid"] => int(1)</pre>

	<pre> ["content"] => string(60) "欢迎使用 ThinsnsV3 , 与同事高效协作 , 快乐沟通。" ["to_uid"] => array(1) { [0] => string(5) "27852" } ["user_info"] => bool(false) } ["content"] => string(60) "欢迎使用 ThinsnsV3 , 与同事高效协作 , 快乐沟通。" } } } </pre>	
字段说明	成功时 :	
	失败时 :	空数组
其他说明 :		

5.3 微博

5.3.1 【全部微博】WeiboStatuses/public_timeline

接口名称 :	WeiboStatuses/public_timeline
接口描述 :	返回公开的全部微博信息
演示地址 :	
输入参数 :	(int) since_id , 起始微博 ID (int) max_id , 最大微博 ID (int) count , 分页显示时 , 指定每页显示条数 (默认 20)

	<p>(int) page , 分页显示时 , 指定获取的页码 (默认取第 1 页)</p> <p>(string) format , 返回格式可选 json/php/test 格式 (默认 json 格式 , 为了阅读方便 , 以下文档中输出结果均以 test 格式显示)</p>
输出结果 :	<pre> 0 => array ('feed_id' => '81' , 'uid' => '14983' , 'content' => 'sssss ' , 'ctime' => '2012-05-30 12:07' , 'from' => '0' , 'type' => 'postimage' , 'has_attach' => 1, 'attach' => array (0 => array ('attach_id' => '247' , 'attach_name' => '433.png' , 'attach_url' => </pre>

```
'http://localhost/work/st/data/upload/2012/0530/12/4fc59cee4154c.
png' ,

),

1 =>

array (

    'attach_id' => '248' ,

    'attach_name' => '450.jpg' ,

    'attach_url' =>

'http://localhost/work/st/data/upload/2012/0530/12/4fc59cf1c7219.j
pg' ,

),

),

'uname' => '杨佳胜' ,

'avatar_big' =>

'http://localhost/work/st/data/upload/avatar/ce/6f/31/big_13383424
81.jpg' ,

'avatar_middle' =>

'http://localhost/work/st/data/upload/avatar/ce/6f/31/middle_13383
42481.jpg' ,

'avatar_small' =>
```

```
'http://localhost/work/st/data/upload/avatar/ce/6f/31/small_133834
2481.jpg' ,

),

1 =>

array (

    'feed_id' => '80' ,

    'uid' => '14983' ,

    'content' => '&nbsp;234234324324' ,

    'ctime' => '2012-05-30 11:16' ,

    'from' => '0' ,

    'type' => 'repost' ,

    'transpond_id' => '78' ,

    'transpond_data' =>

    array (

        'feed_id' => '78' ,

        'uid' => '14983' ,

        'type' => 'postimage' ,

        'app' => 'public' ,

        'app_row_id' => '0' ,

        'publish_time' => '1338347394' ,
```

```

        'is_del' => '0' ,

        'from' => '0' ,

        'comment_count' => '0' ,

        'report_count' => '0' ,

        'client_ip' => '::1' ,

        'ctime' => '1970-01-01 08:00' ,

        'content' =>

        'img{data=[SITE_URL]/addons/theme/stv1/_stastic/image/expression
/miniblog/sad.gif}img{data=[SITE_URL]/addons/theme/stv1/_stastic/i
mage/expression/miniblog/moon.gif}img{data=[SITE_URL]/addons/th
eme/stv1/_stastic/image/expression/miniblog/ok.gif}img{data=[SITE_
URL]/addons/theme/stv1/_stastic/image/expression/miniblog/time.gif
}' ,

        'uname' => '杨佳胜' ,

        'avatar_big' =>

        'http://localhost/work/st/data/upload/avatar/ce/6f/31/big_13383424
81.jpg' ,

        'avatar_middle' =>

        'http://localhost/work/st/data/upload/avatar/ce/6f/31/middle_13383
42481.jpg' ,

```

	<pre> 'avatar_small' => 'http://localhost/work/st/data/upload/avatar/ce/6f/31/small_133834 2481.jpg' ,), 'has_attach' => 0, 'uname' => '杨佳胜' , 'avatar_big' => 'http://localhost/work/st/data/upload/avatar/ce/6f/31/big_13383424 81.jpg' , 'avatar_middle' => 'http://localhost/work/st/data/upload/avatar/ce/6f/31/middle_13383 42481.jpg' , 'avatar_small' => 'http://localhost/work/st/data/upload/avatar/ce/6f/31/small_133834 2481.jpg' ,), </pre>	
字段说明：	成功时：	
	失败时：	
其他说明：		

5.3.2 【关注微博】 WeiboStatuses/friends_timeline

接口名称：	WeiboStatuses/friends_timeline
接口描述：	返回用户所关注用户的最新微博信息
演示地址：	
输入参数：	(int) mid , 当前登录用户 UID (int) since_id , 起始微博 ID (int) max_id , 最大微博 ID (int) count , 分页显示时, 指定每页显示条数 (默认 20) (int) page , 分页显示时, 指定获取的页码 (默认取第 1 页) (string) format , 返回格式可选 json/php/test 格式 (默认 json 格式)
输出结果：	同 public_timeline
字段说明：	同 public_timeline
其他说明：	

5.3.3 【发布微博】 WeiboStatuses/user_timeline

接口名称：	WeiboStatuses/user_timeline
接口描述：	返回用户发布的微博
演示地址：	
输入参数：	(int) user_id , 指定用户 UID (int) since_id , 起始微博 ID

	(int) max_id , 最大微博 ID (int) count , 分页显示时 , 指定每页显示条数 (默认 20) (int) page , 分页显示时 , 指定获取的页码 (默认取第 1 页) (string) format , 返回格式可选 json/php/test 格式 (默认 json 格式)
输出结果 :	同 public_timeline
字段说明 :	同 public_timeline
其他说明 :	

5.3.4 【@我的微博列表】WeiboStatuses/mentions_feed

接口名称 :	WeiboStatuses/mentions_feed
接口描述 :	获取@我的微博列表
演示地址 :	
输入参数 :	(int) mid , 当前登陆用户 UID (int) since_id , 起始微博 ID (int) max_id , 最大微博 ID (int) count , 分页显示时 , 指定每页显示条数 (默认 20) (int) page , 分页显示时 , 指定获取的页码 (默认取第 1 页) (string) format , 返回格式可选 json/php/test 格式 (默认 json 格式)
输出结果 :	
字段说明 :	

其他说明：	
-------	--

5.3.5 【微博详情】WeiboStatuses/show

接口名称：	WeiboStatuses/show
接口描述：	某条详细微博内容
演示地址：	
输入参数：	(int) id , 微博 ID (string) format , 返回格式可选 json/php/test 格式 (默认 json 格式)
输出结果：	
字段说明：	
其他说明：	

5.3.6 【发文字微博】WeiboStatuses/create

接口名称：	WeiboStatuses/create
接口描述：	发布一条新微博
输入参数：	(int) mid , 当前登录用户 UID (string) content , 微博内容 (int) from , 微博来源 (0 : 网站 , 1 : 手机网页版 , 2 : andorid 客户端 , 3 : iphone 客户端) ((string) appname : 微薄所属的应用, 默认 public,如博客就是 blog

	<p>(int) app_id: 微薄在所属的应用中的 ID , 如日志 ID , 默认为空</p> <p>(string) format , 返回格式可选 json/php/test 格式 (默认 json 格式)</p>
输出结果 :	(int) id , 微博 ID (失败返回0 , 成功返回微博 ID)
其他说明 :	<pre> array ('feed_id' => '81', 'uid' => '14983', 'type' => 'postimage', 'app' => 'public', 'app_row_id' => '0', 'publish_time' => '1338350836', 'is_del' => '0', 'from' => '0', 'comment_count' => '2', 'report_count' => '0', 'client_ip' => '::1', 'ctime' => '2012-05-30 12:07', 'content' => 'sssss ', 'uname' => '杨佳胜', 'avatar_big' => 'http://localhost/work/st/data/upload/avatar/ce/6f/31/big_133834248 </pre>

```
1.jpg',
    'avatar_middle' =>
'http://localhost/work/st/data/upload/avatar/ce/6f/31/middle_133834
2481.jpg',
    'avatar_small' =>
'http://localhost/work/st/data/upload/avatar/ce/6f/31/small_13383424
81.jpg',
    'has_attach' => 1,
    'attach' =>
array (
    0 =>
array (
        'attach_id' => '247',
        'attach_name' => '433.png',
        'attach_url' =>
'http://localhost/work/st/data/upload/2012/0530/12/4fc59cee4154c.p
ng',
    ),
    1 =>
array (
```

	<pre> 'attach_id' => '248', 'attach_name' => '450.jpg', 'attach_url' => 'http://localhost/work/st/data/upload/2012/0530/12/4fc59cf1c7219.jpg',),),)</pre>
--	---

5.3.7 【发图片微博】 WeiboStatuses/upload

接口名称：	WeiboStatuses/upload
接口描述：	发布一条带有图片微博
输入参数：	<p>(int) mid , 当前登录用户 UID</p> <p>(string) content , 微博内容</p> <p>(int) from , 微博来源 (0 : 网站 , 1 : 手机网页版 , 2 : andorid 客户端 , 3 : iphone 客户端)</p> <p>(string) appname : 微薄所属的应用 , 默认 public,如博客就是 blog</p> <p>(int) app_id: 微薄在所属的应用中的 ID , 如日志 ID , 默认为空</p> <p>图片变量 : \$_FILES['Filedata']</p> <p>(string) format , 返回格式可选 json/php/test 格式 (默认 json 格式)</p>

输出结果：	(int) id , 微博 ID (失败返回0 , 成功返回微博 ID)
其他说明：	

5.3.8 【删除微博】WeiboStatuses/destroy

接口名称：	WeiboStatuses/destroy
接口描述：	删除微博
演示地址：	
输入参数：	(int) id , 指定的微博 ID (int) mid , 当前登录用户 UID (string) format , 返回格式可选 json/php/test 格式 (默认 json 格式)
输出结果：	(int) 1/0 , 成功返回1 , 失败为0
字段说明：	
其他说明：	

5.3.9 【转发微博】WeiboStatuses/repost

接口名称：	WeiboStatuses/repost
接口描述：	转发一条微博
输入参数：	(int) id , 指定的微博 ID (string) app_name : 所转发的内容的应用名称 , 默认 public (string)content , 微博内容 (string) type:被转发的内容所在的表 , 默认 feed

	(int)comment , 0:不评论给原作者 , 1 评论给原作者 (string) format , 返回格式可选 json/php/test 格式 (默认 json 格式)
输出结果 :	(int) 微博 ID , 成功返回微博 ID , 失败为0
其他说明 :	

5.3.10 【评论列表】WeiboStatuses/comments

接口名称 :	WeiboStatuses/comments
接口描述 :	返回某条微博的评论列表
演示地址 :	
输入参数 :	(int)id , 微博 ID (int) since_id , 起始评论 ID (int) max_id , 最大回复 ID (int) count , 分页显示时 , 指定每页显示条数 (默认 20) (int) page , 分页显示时 , 指定获取的页码 (默认取第 1 页) (string) format , 返回格式可选 json/php/test 格式 (默认 json 格式)
输出结果 :	<pre> 0 => array ('comment_id' => '138', 'app' => 'public', </pre>

```
'table' => 'feed',

'row_id' => '86',

'app_uid' => '14983',

'uid' => '14983',

'content' =>

'123213img{data=http://localhost/work/st/addons/theme/stv1/_stasti
c/image/expression/miniblog/cry.gif}',

'to_comment_id' => '0',

'to_uid' => '0',

'data' => 'N;',

'ctime' => '1338374464',

'is_del' => '0',

'client_type' => '0',

'user_info' =>

array (

    'uid' => '14983',

    'login' => 'yangjs17@yeah.net',

    'login_salt' => '11111',

    'uname' => '杨佳胜',

    'email' => 'yangjs17@yeah.net',
```

```
'sex' => '2',

'location' => '浙江 丽水市',

'is_audit' => '0',

'is_active' => '1',

'is_init' => '1',

'ctime' => '1302696447',

'identity' => '1',

'api_key' => NULL,

'domain' => 'ilover17',

'province' => '943',

'city' => '0',

'area' => '0',

'reg_ip' => '127.0.0.1',

'lang' => 'zh-cn',

'timezone' => 'PRC',

'department_id' => '2',

'avatar_big' =>

'http://localhost/work/st/data/upload/avatar/ce/6f/31/big_133837216

6.jpg',

'avatar_middle' =>
```

	<pre> 'http://localhost/work/st/data/upload/avatar/ce/6f/31/middle_133837 2166.jpg', 'avatar_small' => 'http://localhost/work/st/data/upload/avatar/ce/6f/31/small_13383721 66.jpg', 'avatar_url' => 'http://localhost/work/st/index.php?app=public&mod=Attach&act=av atar&uid=14983', 'space_url' => 'http://localhost/work/st/ilover17', 'space_link' => '杨佳胜',),), </pre>
其他说明：	

5.3.11 【评论我的】WeiboStatuses/comments_to_me

接口名称：	WeiboStatuses/comments_to_me
接口描述：	返回我收到的评论列表
演示地址：	

输入参数：	<p>(int)uid , 当前用户 ID</p> <p>(int) since_id , 起始评论 ID</p> <p>(int) max_id , 最大回复 ID</p> <p>(int) count , 分页显示时 , 指定每页显示条数 (默认 20)</p> <p>(int) page , 分页显示时 , 指定获取的页码 (默认取第 1 页)</p> <p>(string) format , 返回格式可选 json/php/test 格式 (默认 json 格式)</p>
输出结果：	<pre> array (0 => array ('comment_id' => '138', 'app' => 'public', 'table' => 'feed', 'row_id' => '86', 'app_uid' => '14983', 'uid' => '14983', 'content' => '123213img{data=http://localhost/work/st/addons/theme/stv1/_stasti c/image/expression/miniblog/cry.gif}', 'to_comment_id' => '0', 'to_uid' => '0',)) </pre>

```
'data' => 'N;',  
  
'ctime' => '1338374464',  
  
'is_del' => '0',  
  
'client_type' => '0',  
  
'user_info' =>  
array (  
    'uid' => '14983',  
    'login' => 'yangjs17@yeah.net',  
    'login_salt' => '11111',  
    'uname' => '杨佳胜',  
    'email' => 'yangjs17@yeah.net',  
    'sex' => '2',  
    'location' => '浙江 丽水市',  
    'is_audit' => '0',  
    'is_active' => '1',  
    'is_init' => '1',  
    'ctime' => '1302696447',  
    'identity' => '1',  
    'api_key' => NULL,  
    'domain' => 'ilover17',
```

```
'province' => '943',  
  
'city' => '0',  
  
'area' => '0',  
  
'reg_ip' => '127.0.0.1',  
  
'lang' => 'zh-cn',  
  
'timezone' => 'PRC',  
  
'department_id' => '2',  
  
'avatar_big' =>  
'http://localhost/work/st/data/upload/avatar/ce/6f/31/big_133837216  
6.jpg',  
  
'avatar_middle' =>  
'http://localhost/work/st/data/upload/avatar/ce/6f/31/middle_133837  
2166.jpg',  
  
'avatar_small' =>  
'http://localhost/work/st/data/upload/avatar/ce/6f/31/small_13383721  
66.jpg',  
  
'avatar_url' =>  
'http://localhost/work/st/index.php?app=public&mod=Attach&act=av  
atar&uid=14983',  
  
'space_url' => 'http://localhost/work/st/ilover17',
```

```
'space_link' => '<a href=\'http://localhost/work/st/ilover17\'
title=\'杨佳胜\' target=\'_blank\' uid=\'14983\'
event-node=\'face_card\'>杨佳胜</a>',

),

'sourceInfo' =>

array (

    'uid' => '14983',

    'source_user' => '<a href=\'http://localhost/work/st/ilover17\'
title=\'杨佳胜\' target=\'_blank\' uid=\'14983\'
event-node=\'face_card\'>杨佳胜</a>',

    'source_user_info' =>

        array (

            'uid' => '14983',

            'login' => 'yangjs17@yeah.net',

            'login_salt' => '11111',

            'uname' => '杨佳胜',

            'email' => 'yangjs17@yeah.net',

            'sex' => '2',

            'location' => '浙江 丽水市',

            'is_audit' => '0',
```

```
'is_active' => '1',  
  
'is_init' => '1',  
  
'ctime' => '1302696447',  
  
'identity' => '1',  
  
'api_key' => NULL,  
  
'domain' => 'ilover17',  
  
'province' => '943',  
  
'city' => '0',  
  
'area' => '0',  
  
'reg_ip' => '127.0.0.1',  
  
'lang' => 'zh-cn',  
  
'timezone' => 'PRC',  
  
'department_id' => '2',  
  
'avatar_big' =>  
'http://localhost/work/st/data/upload/avatar/ce/6f/31/big_133837216  
6.jpg',  
  
'avatar_middle' =>  
'http://localhost/work/st/data/upload/avatar/ce/6f/31/middle_133837  
2166.jpg',  
  
'avatar_small' =>
```

```
'http://localhost/work/st/data/upload/avatar/ce/6f/31/small_13383721
66.jpg',

    'avatar_url' =>

'http://localhost/work/st/index.php?app=public&mod=Attach&act=av
atar&uid=14983',

    'space_url' => 'http://localhost/work/st/ilover17',

    'space_link' => '<a href=\'http://localhost/work/st/ilover17\'
title=\'杨佳胜\' target=\'_blank\' uid=\'14983\'
event-node=\'face_card\'>杨佳胜</a>',

    ),

    'source_type' => '微博',

    'source_title' => '',

    'source_url' =>

'http://localhost/work/st/index.php?app=public&mod=Profile&act=fe
ed&feed_id=86&uid=14983',

    'source_content' => '&nbsp;dwadawd',

    'ctime' => '1338367381',

    'app' => 'public',

    'source_table' => 'feed',

    'source_id' => '86',
```

),)
字段说明：	
其他说明：	

5.3.12 【我评论的】WeiboStatuses/comments_by_me

接口名称：	WeiboStatuses/comments_by_me
接口描述：	返回我收到的评论列表
输入参数：	(int)uid，当前用户 ID (int) since_id，起始评论 ID (int) max_id，最大回复 ID (int) count，分页显示时，指定每页显示条数（默认 20） (int) page，分页显示时，指定获取的页码（默认取第 1 页） (string) format，返回格式可选 json/php/test 格式（默认 json 格式）
输出结果：	同：public_timeline
其他说明：	

5.3.13 【发出评论】WeiboStatuses/comment

接口名称：	WeiboStatuses/comment
接口描述：	发出评论
演示地址：	

输入参数：	<p>app_name 评论所在的应用名称，微薄默认未 public</p> <p>app_row_id 评论的内容在应用中的 ID，默认为 0 (如转发的原微薄 ID)</p> <p>app_uid 评论的内容的作者 ID (如转发时原微薄的作者)</p> <p>comment_old 是否评论给原文作者</p> <p>content 评论内容 (必须)</p> <p>ifShareFeed 是否分享到微薄</p> <p>row_id 评论的对象 ID，如评论微薄就是 feed_id</p> <p>table_name 评论的对象的表名，如微薄里评论就是 feed</p> <p>to_comment_id 回复的评论的 ID</p> <p>to_uid 回复的人的 uid</p> <p>(string) format，返回格式可选 json/php/test 格式 (默认 json 格式)</p>
输出结果：	成功返回1，失败返回0
其他说明：	

5.3.14 【收藏微博列表】WeiboStatuses/favorite_feed

接口名称：	WeiboStatuses/favorite_feed
接口描述：	返回用户的收藏的微博 (feed) 列表
输入参数：	<p>(int)mid，当前用户 ID</p> <p>(int) since_id，起始评论 ID</p> <p>(int) max_id，最大回复 ID</p>

	<p>(int) count , 分页显示时 , 指定每页显示条数 (默认 20)</p> <p>(int) page , 分页显示时 , 指定获取的页码 (默认取第 1 页)</p> <p>(string) format , 返回格式可选 json/php/test 格式 (默认 json 格式)</p>
输出结果	<p>同 public_timeline 一致</p> <pre>array (0 => array ('feed_id' => '174', 'uid' => '14983', 'type' => 'repost', 'app' => 'public', 'app_row_table' => 'feed', 'app_row_id' => '173', 'publish_time' => '1338780126', 'is_del' => '0', 'from' => '0', 'comment_count' => '0', 'repost_count' => '0', 'client_ip' => '::1', 'ctime' => '2012-06-04 11:22', 'content' => '娃打我的娃', 'uname' => '杨佳胜', 'avatar_big' => 'http://192.168.1.133/work/st/data/upload/avatar/ce/6f/31/big_1338876725.jpg', 'avatar_middle' => 'http://192.168.1.133/work/st/data/upload/avatar/ce/6f/31/middle_1338876725.jpg', 'avatar_small' => 'http://192.168.1.133/work/st/data/upload/avatar/ce/6f/31/small_1338876725.jpg', 'transpond_id' => '173', 'transpond_data' => array ('feed_id' => '173', 'uid' => '27851',</pre>

	<pre> 'type' => 'post', 'app' => 'public', 'app_row_table' => 'feed', 'app_row_id' => '0', 'publish_time' => '1338780099', 'is_del' => '0', 'from' => '0', 'comment_count' => '0', 'repost_count' => '0', 'client_ip' => '192.168.1.131', 'ctime' => '2012-06-04 11:21', 'content' => ' @杨佳胜 ', 'uname' => 'yangjs', 'avatar_big' => 'http://192.168.1.133/work/st/data/upload/avatar/14/1f/ac/big_13388 76725.jpg', 'avatar_middle' => 'http://192.168.1.133/work/st/data/upload/avatar/14/1f/ac/middle_13 38876725.jpg', 'avatar_small' => 'http://192.168.1.133/work/st/data/upload/avatar/14/1f/ac/small_133 8876725.jpg', 'has_attach' => 0, 'iscoll' => '',), 'has_attach' => 0, 'iscoll' => array ('source_table_name' => 'feed', 'source_id' => 174, 'source_app' => 'public', 'uid' => 27847, 'ctime' => 1338881169, 'collection_id' => 64,),), </pre>
--	---

5.3.15 【添加收藏】WeiboStatuses/favorite_create

接口名称：	WeiboStatuses/favorite_create
接口描述：	添加收藏
输入参数：	(int)uid，当前用户 ID (int) source_table_name，收藏的资源表名，如微薄是 feed (int) source_id 资源 ID，如微薄 ID (int) source_app 资源所在的应用，如微薄默认为 public (string) format，返回格式可选 json/php/test 格式（默认 json 格式）
输出结果：	(int) 1/0，1表示成功，0表示失败
其他说明：	

5.3.16 【取消收藏】WeiboStatuses/favorite_destroy

接口名称：	WeiboStatuses/favorite_destroy
接口描述：	取消收藏
演示地址：	
输入参数：	(int) source_id，资源 ID，如微薄的 feed_id (int) source_table_name 资源表名，如 feed (string) format，返回格式可选 json/php/test 格式（默认 json 格式）
输出结果：	(int) 1/0，1表示成功，0表示失败
其他说明：	

5.3.17 【收藏的话题列表】WeiboStatuses/weibo_topices

接口名称：	WeiboStatuses/weibo_topices
接口描述：	微博话题列表
输入参数：	(int) topic_id , 话题 ID (string)data['content'] , 当前话题内容 (int) since_id , 起始评论 ID (int) max_id , 最大回复 ID (int) count , 分页显示时 , 指定每页显示条数 (默认 20) (int) page , 分页显示时 , 指定获取的页码 (默认取第 1 页) (string) format , 返回格式可选 json/php/test 格式 (默认 json 格式)
输出结果：	同 : public_timeline
其他说明：	

5.3.18 【关注话题】WeiboStatuses/weibo_topices_create

接口名称：	WeiboStatuses/weibo_topices_create
接口描述：	关注话题
输入参数：	(int) topic_id , 话题 ID (string)data['content'] , 当前话题内容 (string) format , 返回格式可选 json/php/test 格式 (默认 json 格式)
输出结果：	(int) 1/0 , 成功为1 , 失败为0

其他说明：	
-------	--

5.3.19 【取消关注】WeiboStatuses/weibo_topices_destroy

接口名称：	WeiboStatuses/weibo_topices_destroy
接口描述：	取消关注话题
输入参数：	(int)topic_id , 话题 ID (string)data["content "], 当前话题内容 (string) format , 返回格式可选 json/php/test 格式 (默认 json 格式)
输出结果：	(int) 1/0 , 成功为1 , 失败为0
其他说明：	

5.3.20 【搜索微博】WeiboStatuses/weibo_search_weibo

接口名称：	WeiboStatuses/weibo_search_weibo
接口描述：	根据关键字搜索微博
演示地址：	
输入参数：	(string) key , 查询关键词 (int) max_id , 最大微博 ID (int) count , 分页显示时 , 指定每页显示条数 (默认 20) (int) page , 分页显示时 , 指定获取的页码 (默认取第 1 页) (string) format , 返回格式可选 json/php/test 格式 (默认 json 格式)
输出结果：	同 public_timeline

其他说明：	
-------	--

5.3.21 【搜索用户】WeiboStatuses/weibo_search_user

接口名称：	WeiboStatuses/weibo_search_user
接口描述：	根据关键字搜索用户
演示地址：	
输入参数：	(string) key , 查询关键词 (int) max_id , 最大用户 ID (int) count , 分页显示时, 指定每页显示条数 (默认 20) (int) page , 分页显示时, 指定获取的页码 (默认取第 1 页) (string) format , 返回格式可选 json/php/test 格式 (默认 json 格式)
输出结果：	同关注、粉丝列表
字段说明：	
其他说明：	

5.4 用户

5.4.1 【用户信息】User/show

接口名称：	User/show
接口描述：	根据传入的用户 ID 返回用户信息
演示地址：	

<p>输入参数：</p>	<p>(int) user_id , 用户 UID,如果无 , 默认为当前登陆用户 UID</p> <p>(string)user_name , 用户昵称 , (user_id 或 user_name,两者必选一)</p> <p>(string) format , 返回格式可选 json/php/test 格式 (默认 json 格式 , 为了阅读方便 , 以下文档中输出结果均以 test 格式显示)</p>
<p>输出结果：</p>	<pre>'uid' => '14983', 'login' => 'yangjs17@yeah.net', 'login_salt' => '11111', 'uname' => '阳阳', 'email' => 'yangjs17@yeah.net', 'sex' => '男', 'location' => '湖南 长沙', 'is_audit' => '0', 'is_active' => '1', 'is_followed' => 'eachfollow', 'is_init' => '1', 'ctime' => '1302696447', 'identity' => '1', 'api_key' => NULL, 'domain' => 'ilover17', 'province' => '943', 'city' => '0', 'area' => '0', 'reg_ip' => '127.0.0.1', 'lang' => 'zh-cn', 'timezone' => 'PRC', 'department_id' => '2', 'avatar_big' 'http://dev.zhishisoft.com/st/data/upload/avatar/ce/6f/31/big_1338177747.jpg', 'avatar_middle' 'http://dev.zhishisoft.com/st/data/upload/avatar/ce/6f/31/middle_1338177747.jpg',</pre>

	<pre> 'avatar_small' 'http://dev.zhishisoft.com/st/data/upload/avatar/ce/6f/31/small_1338177747.jpg', 'avatar_url' 'http://dev.zhishisoft.com/st/index.php?app=public&mod=Attach&act=avatar&uid=14983', 'space_url' => 'http://dev.zhishisoft.com/st/ilover17', 'space_link' => '\&\&\&', 'department' => array ('dept_id' => '2', 'title' => '\&\&B', 'parent_dept_id' => '0', 'display_order' => '2', 'ctime' => NULL,), 'profile' => array ('intro' => array ('name' => '\&\&\&\&', 'value' => '213',), 'work_position' => array ('name' => '\&\&', 'value' => '',), 'work_project' => array ('name' => '\&\&\&\&', 'value' => '',), 'work_director' => array (</pre>	<pre> => => </pre>
--	---	--------------------------

	<pre> 'name' => '????', 'value' => '10336,10000',),), 'count_info' => array ('favorite_count' => 0, 'feed_count' => 33, 'follower_count' => 6, 'following_count' => 14, 'unread_atme' => 0, 'unread_comment' => 0, 'weibo_count' => 54,), 'user_tag' => '??、??、??、??', 'follow_state' => array ('following' => 0, 'follower' => 0,), 'last_feed' => array ('feed_id' => '310', 'uid' => '14983', 'type' => 'post', 'app' => 'public', 'app_row_id' => '0', 'publish_time' => '1338136654', 'is_del' => '0', 'from' => '0', 'comment_count' => '0', 'report_count' => '0', 'client_ip' => '114.246.94.151', 'feed_data' => 'a:9:{s:7:"content";s:0:"";s:4:"body";s:333:" @{{uid=10336}} @{{uid=27861}} @{{uid=10045}} @{{uid=27850}} </pre>
--	---

	<pre> ThinsnsV3: 、 、 (feed) 、 ThinsnsV3、 ThinsnsV3BUG, review () 。 "s:9:"attach_id";s:0:"";s:3:"uid";i:14983;s:3:"app";s:6:"public"; s:4:"type";s:4:"post";s:10:"app_row_id";i:0;s:12:"publish_time";i:1338136654;s:4:" from";s:1:"0";}', 'info' => ' ', 'title' => ' ', 'body' => ' @ &nbsp; @ &nbsp; @ &nbsp; @ &nbsp; ThinsnsV3: 、 、 (feed) 、 ThinsnsV3、 ThinsnsV3BUG, review () 。 ', 'sbody' => ', 'user_info' => array ('uid' => '14983', 'login' => 'yangjs17@yeah.net', 'login_salt' => '11111', 'uname' => ' ', 'email' => 'yangjs17@yeah.net', 'sex' => '2', 'location' => ' ', 'is_audit' => '0', 'is_active' => '1', 'is_init' => '1', 'ctime' => '1302696447', 'identity' => '1', 'api_key' => NULL, 'domain' => 'ilover17', </pre>
--	---

	<pre>'province' => '943', 'city' => '0', 'area' => '0', 'reg_ip' => '127.0.0.1', 'lang' => 'zh-cn', 'timezone' => 'PRC', 'department_id' => '2', 'avatar_big' => 'http://dev.zhishisoft.com/st/data/upload/avatar/ce/6f/31/big_1338177747.jpg', 'avatar_middle' => 'http://dev.zhishisoft.com/st/data/upload/avatar/ce/6f/31/middle_1338177747.jpg', 'avatar_small' => 'http://dev.zhishisoft.com/st/data/upload/avatar/ce/6f/31/small_1338177747.jpg', 'avatar_url' => 'http://dev.zhishisoft.com/st/index.php?app=public&mod=Attach&act=avatar&uid=14983', 'space_url' => 'http://dev.zhishisoft.com/st/ilover17', 'space_link' => '\',), 'actions' => array ('comment' => 'true', 'repost' => 'true', 'favor' => 'true', 'report' => 'true', 'delete' => 'true',),</pre>	
字段说明：	成功时：	

	失败时:	
其他说明：	以上只为格式，具体字段跟数据由服务端调配	

5.4.2 【上传头像】User/upload_face

接口名称：	User/upload_face	
接口描述：	上传用户头像	
演示地址：		
输入参数：	(bin) multipart form data , (Files) Filedata 通过 POST 方式发送图片二进制数据（参考微博上传图片）传入的都跟变量（必填） (int) mid，当前登录用户 UID (string) format，返回格式可选 json/php/test 格式（默认 json 格式，为了阅读方便，以下文档中输出结果均以 test 格式显示）	
输出结果：	(int) 整数	
字段说明：	成功时：	
	失败时：	
其他说明：		

5.4.3 【添加关注】User/follow_create

接口名称：	User/follow_create	
接口描述：	关注用户	
演示地址：		

输入参数：	(int) mid , 当前用户 ID (选填) (int) user_id , 关注用户 ID (必填) (string) format , 返回格式可选 json/php/test 格式 (默认 json 格式 , 为了阅读方便 , 以下文档中输出结果均以 test 格式显示)	
输出结果：	array ('following' => 1, 'follower' => 0,)	
字段说明	成功时：	following : 1 表示已关注用户 , 0 表示没有关注 follower : 1 表示被关注 , 0 表示否
	失败时：	0
其他说明：	Following 表示是否关注 , follower 表示是否被关注	

5.4.4 【取消关注】User/follow_destroy

接口名称：	User/follow_destroy
接口描述：	取消关注用户
演示地址：	
输入参数：	(int) uid , 当前用户 ID //当前 mid 不需要传入 (int) user_id , 关注用户 ID (必填) (string) format , 返回格式可选 json/php/test 格式 (默认 json 格式 , 为了阅读方便 , 以下文档中输出结果均以 test 格式显示)
输出结果	操作成功输出： array ('following' => 1, 'follower' => 0,)

字段说明：	成功时：	following : 1 表示已关注用户，0 表示没有关注 follower : 1 表示被关注，0 表示否
	失败时：	0
其他说明：	Following 表示是否关注，follower 表示是否被关注	

5.4.5 【粉丝列表】User/user_followers

接口名称：	User/user_followers
接口描述：	返回用户的粉丝列表
演示地址：	
输入参数：	<p>(int) user_id, 查看的用户 ID (默认当前用户 ID)</p> <p>(int) since_id, 起始关联 ID</p> <p>(int) max_id, 最大关联 ID</p> <p>(int) count, 分页显示时, 指定每页显示条数 (默认 20)</p> <p>(int) page, 分页显示时, 指定获取的页码 (默认取第 1 页)</p> <p>(string) format, 返回格式可选 json/php/test 格式 (默认 json 格式, 为了阅读方便, 以下文档中输出结果均以 test 格式显示)</p>
输出结果：	<pre>array (0 => array ('follow_id' => '59' ,</pre>

```
'uid' => '27847' ,

'remark' => '2222' , -- 备注

'department' =>

array (

    'dept_id' => '7' ,

    'title' => '部门 A-1-1' ,

    'parent_dept_id' => '3' ,

    'display_order' => '1' ,

    'ctime' => NULL,

),

'follow_state' =>

array (

    'following' => 0,

    'follower' => 1,

),

'profile' =>

array (

    'intro' =>

        array (

            'name' => '个人简介' ,
```

```
        'value' => 'wawd' ,  
    ),  
    'birthday' =>  
    array (  
        'name' => '生日' ,  
        'value' => '1335974400' ,  
    ),  
    'work_experience' =>  
    array (  
        'name' => '工作经历' ,  
        'value' => 'wawaw' ,  
    ),  
    'work_position' =>  
    array (  
        'name' => '职务' ,  
        'value' => 'sss' ,  
    ),  
    'work_project' =>  
    array (  
        'name' => '项目经验' ,
```


	<pre> 'value' => 'awdawd' ,), 'work_director' => array ('name' => '直接主管' , 'value' => '14983' ,),), 'avatar_big' => 'http://localhost/work/st/data/upload/avatar/37/08/d4/big_1338261 138.jpg' , 'avatar_middle' => 'http://localhost/work/st/data/upload/avatar/37/08/d4/middle_1338 261138.jpg' , 'avatar_small' => 'http://localhost/work/st/data/upload/avatar/37/08/d4/small_133826 1138.jpg' ,), </pre>	
字段说明：	成功时：	

	失败时：	
其他说明：		

5.4.6 【关注列表】User/user_following

接口名称：	User/user_followers
接口描述：	返回用户的关注列表
演示地址：	
输入参数：	<p>(int) uid , 当前用户 ID</p> <p>(int) since_id , 起始用户 ID</p> <p>(int) max_id , 最大用户 ID</p> <p>(int) count , 分页显示时 , 指定每页显示条数 (默认 20)</p> <p>(int) page , 分页显示时 , 指定获取的页码 (默认取第 1 页)</p> <p>(string) format , 返回格式可选 json/php/test 格式 (默认 json 格式 , 为了阅读方便 , 以下文档中输出结果均以 test 格式显示)</p>
输出结果：	<pre>array (0 => array ('follow_id' => '77', 'uid' => '10000', 'remark' => "", 'department' => array ('dept_id' => '1', 'title' => '部门 A', 'parent_dept_id' => '0', 'display_order' => '1',</pre>

	<pre> 'ctime' => NULL,), 'follow_state' => array ('following' => 1, 'follower' => 1,), 'profile' => array (), 'avatar_big' => 'http://localhost/work/st/addons/theme/stv1/_stastic/image/noavatar/big.jpg', 'avatar_middle' => 'http://localhost/work/st/addons/theme/stv1/_stastic/image/noavatar/middle.jpg', 'avatar_small' => 'http://localhost/work/st/addons/theme/stv1/_stastic/image/noavatar/small.jpg',), 1 => array ('follow_id' => '80', 'uid' => '10008', 'remark' => 'ers', 'department' => array ('dept_id' => '1', 'title' => '部门 A', 'parent_dept_id' => '0', 'display_order' => '1', 'ctime' => NULL,), 'follow_state' => array ('following' => 1, 'follower' => 0,), 'profile' => array (), 'avatar_big' => 'http://localhost/work/st/addons/theme/stv1/_stastic/image/noavatar/big.jpg', </pre>
--	---

	<pre>'avatar_middle' => 'http://localhost/work/st/addons/theme/stv1/_stastic/image/noavatar/middle.jpg', 'avatar_small' => 'http://localhost/work/st/addons/theme/stv1/_stastic/image/noavatar/small.jpg',)</pre>
其他说明：	

5.5 聊天

5.5.1 【对话列表】 Message/get_message_list

接口名称：	Message/get_message_list
接口描述：	返回当前用户的所有对话列表，倒序（原来 API 中的 BOX 函数）
演示地址：	http://192.168.1.200/ThinsnsV3/index.php?app=api&mod=AppList&act=get_message_list&oauth_token=1e52300bf6576b2efe63a7fa4ec37b9c&oauth_token_secret=cb730170da8c17dc7f5683669303a09f&format=json
输入参数：	<p>(int) mid，当前登录用户 UID（选填，默认当前用户）</p> <p>(int) since_id，起始对话列表 ID（选填）</p> <p>(int) max_id，最大对话列表 ID（选填）</p> <p>(int) count，分页显示时，指定每页显示条数（选填，默认 20）</p> <p>(int) page，分页显示时，指定获取的页码（选填，默认取第 1 页）</p> <p>(string) format，返回格式可选 json/php/test 格式（默认 json 格式）</p>
输出结果：	<pre>rray(2) { [0] => array(17) { ["list_id"] => string(1) "6" ["member_uid"] => string(5) "10665" ["new"] => string(1) "0" ["message_num"] => string(1) "1"</pre>

```

["ctime"] => string(16) "2011-12-29 18:15"
["list_ctime"] => string(10) "1325153706"
["from_uid"] => int(10665)
["type"] => string(1) "1"
["title"] => string(11) "new message"
["member_num"] => string(1) "2"
["min_max"] => string(11) "10665_27714"
["mtime"] => string(10) "1325153706"
["last_message"] => array(3) {
    ["from_uid"] => int(10665)
    ["content"] => string(4) "7***"
    ["to_uid"] => array(1) {
        [0] => string(5) "27714"
    }
}
["content"] => string(4) "7***"
["from_uname"] => string(8) "_Lucifer"
["from_face"] => string(80)
"http://localhost/ts/public/themes/newstyle/images/user_pic_middle.gif"
["timestmap"] => string(10) "1325153706"
}
[1] => array(17) {
    ["list_id"] => string(1) "5"
    ["member_uid"] => string(5) "10665"
    ["new"] => string(1) "0"
    ["message_num"] => string(1) "1"
    ["ctime"] => string(16) "2011-12-29 18:10"
    ["list_ctime"] => string(10) "1325153422"
    ["from_uid"] => int(10304)
    ["type"] => string(1) "1"
    ["title"] => string(5) "hello"
    ["member_num"] => string(1) "2"
    ["min_max"] => string(11) "10304_10665"
    ["mtime"] => string(10) "1325153422"
    ["last_message"] => array(3) {
        ["from_uid"] => int(10304)
        ["content"] => string(5) "hello"
        ["to_uid"] => array(1) {
            [0] => string(5) "10665"
        }
    }
}

```

	<pre> ["content"] => string(5) "hello" ["from_uname"] => string(15) "一升的眼泪" ["from_face"] => string(80) "http://localhost/ts/public/themes/newstyle/images/user_pic_middle.gif" ["timestmap"] => string(10) "1325153422" } }</pre>	
字段说明	成功时：	
	失败时：	空数组
其他说明：		

5.5.2 【对话详情】 Message/get_message_detail

接口名称：	Message/get_message_detail
接口描述：	返回指定对话的详细对话
演示地址：	http://192.168.1.200/ThinsnsV3/index.php?app=api&mod=AppList&act=get_message_detail&oauth_token=1e52300bf6576b2efe63a7fa4ec37b9c&oauth_token_secret=cb730170da8c17dc7f5683669303a09f&format=json
输入参数：	<p>(int) mid , 当前登录用户 UID (选填 , 默认当前用户)</p> <p>(int) \$data['id'] , 要显示的消息 ID (必填)</p> <p>(int) since_id , 起始对话 ID (选填)</p> <p>(int) max_id , 最大对话 ID (选填)</p> <p>(int) count , 分页显示时 , 指定每页显示条数 (选填 , 默认 20)</p> <p>(int) page , 分页显示时 , 指定获取的页码 (选填 , 默认取第 1 页)</p> <p>(string) format , 返回格式可选 json/php/test 格式 (默认 json 格式)</p>

输出结果：	<pre> array(1) { [0] => array(10) { ["message_id"] => string(2) "13" ["list_id"] => string(1) "6" ["from_uid"] => string(5) "10665" ["content"] => string(4) "7***" ["is_del"] => string(1) "0" ["mtime"] => string(10) "1325153706" ["from_uname"] => string(8) "_Lucifer" ["from_face"] => string(80) "http://dev.ThinsnsV3Team.com/ts/2.0/public/themes/newstyle/images/user_pic_middl e.gif" ["timestmap"] => string(10) "1325153706" ["ctime"] => string(16) "2011-12-29 18:15" } } </pre>	
字段说明	成功时：	
	失败时：	空数组
其他说明：		

5.5.3 【发布消息】Message/create

接口名称：	Message/create
接口描述：	发布消息
演示地址：	
输入参数：	(int) mid，当前登录用户 UID（选题，默认当前用户） (int)to_uid，发送对象（必填，用户 UID） (string)data['content']，消息内容（必填，消息内容）
输出结果：	(int) 整数

字段说明：	成功时：	MessageID
	失败时：	0
其他说明：		

5.5.4 【回复消息】 Message/reply

接口名称：	Message/reply	
接口描述：	回复某条消息	
演示地址：		
输入参数：	(int) mid，当前登录用户 UID（选题） (int) \$data['id']，回复某个对话 ID（必填） (string) \$data['content']，回复的消息内容（必填） (string) format，返回格式可选 json/php/test 格式（默认 json 格式）	
输出结果：	(int) 整数	
字段说明：	成功时：	messageID
	失败时：	0
其他说明：		

5.5.5 【删除消息】 Message/destroy

接口名称：	Message/destroy	
接口描述：	删除某个对话	
演示地址：		

输入参数：	(int) mid , 当前登录用户 UID (int) \$data['list_id '], 对话列表 ID (必填) (string) format , 返回格式可选 json/php/test 格式 (默认 json 格式 , 为了阅读方便 , 以下文档中输出结果均以 test 格式显示)	
输出结果：	(int)整数	
字段说明：	成功时：	1
	失败时：	0
其他说明：		

5.6 频道

5.6.1 【频道列表】Channel/get_all_channel

接口名称：	Channel/get_all_channel	
接口描述：	获取所有频道列表	
演示地址：		
输入参数：	(int) mid , 用户 ID (选填 , 默认当前用户 ID) (int) since_id , 起始问题 ID (int) max_id , 最大问题 ID (int) count , 分页显示时 , 指定每页显示条数 (默认 20) (int) page , 分页显示时 , 指定获取的页码 (默认取第 1 页) (string) format , 返回格式可选 json/php/test 格式 (默认 json 格式)	

输出结果：		
字段说明：	成功时：	
	失败时：	
其他说明：		

5.6.2 【频道微博】Channel/get_channel_feed

接口名称：	Channel/get_channel_feed	
接口描述：	获取频道微博	
演示地址：		
输入参数：	(int) mid，用户 ID（选填，默认当前用户 ID） (int) since_id，起始问题 ID (int) max_id，最大问题 ID (int) count，分页显示时，指定每页显示条数（默认 20） (int) page，分页显示时，指定获取的页码（默认取第 1 页） (string) format，返回格式可选 json/php/test 格式（默认 json 格式）	
输出结果：		
字段说明：	成功时：	
	失败时：	
其他说明：		

5.7 微吧

5.7.1 【微吧列表】Weiba/get_all_weiba

接口名称：	Weiba/get_weibas	
接口描述：	获取微吧列表	
演示地址：		
输入参数：	(int) user_id , 用户 ID (选填 , 默认当前用户 ID) (int) count , 分页显示时 , 指定每页显示条数 (默认 20) (int) page , 分页显示时 , 指定获取的页码 (默认取第 1 页) (string) format , 返回格式可选 json/php/test 格式 (默认 json 格式)	
输出结果：		
字段说明：	成功时：	Json 数组
	失败时：	
其他说明：		

5.7.2 【关注微吧】Weiba/create

接口名称：	Weiba/create	
接口描述：	关注微吧	
演示地址：		
输入参数：	(int) id , 微吧 ID (string) user_id , 用户 ID (选填 , 默认当前用户 ID)	

	(string) format , 返回格式可选 json/php/test 格式 (默认 json 格式)	
输出结果 :		
字段说明 :	成功时 :	1
	失败时 :	0
其他说明 :		

5.7.3 【取消关注】Weiba/destroy

接口名称 :	Weiba/destroy	
接口描述 :	取消关注的微吧	
接口编号 :		
演示地址 :		
输入参数 :	(int) id , 微吧 ID (string) user_id , 用户 ID (选填 , 默认当前用户 ID) (string) format , 返回格式可选 json/php/test 格式 (默认 json 格式)	
输出结果 :		
字段说明 :	成功时 :	1
	失败时 :	0
其他说明 :		

5.7.4 【帖子列表】Weiba/get_posts

接口名称 :	Weiba/get_posts
--------	-----------------

接口描述：	获取帖子列表	
演示地址：		
输入参数：	(int) id , 微吧 ID (int) count , 分页显示时, 指定每页显示条数 (默认 20) (int) page , 分页显示时, 指定获取的页码 (默认取第 1 页) (string) format , 返回格式可选 json/php/test 格式 (默认 json 格式)	
输出结果：		
字段说明：	成功时：	Json 数组
	失败时：	
其他说明：		

5.7.5 【发布帖子】Weiba/create_post

接口名称：	Weiba/create_post	
接口描述：	发布帖子	
演示地址：		
输入参数：	(int) id , 微吧 ID (string) user_id , 用户 ID (选填, 默认当前用户 ID) (string) title , 帖子标题 (string) content , 帖子内容	
输出结果：		

字段说明：	成功时：	1
	失败时：	0
其他说明：		

5.7.6 【帖子详情】Weiba/post_detail

接口名称：	Weiba/post_detail	
接口描述：	获取帖子详情	
演示地址：		
输入参数：	(int) id , 帖子 ID (string) user_id , 用户 ID (选填 , 默认当前用户 ID)	
输出结果：		
字段说明：	成功时：	
	失败时：	
其他说明：		

5.7.7 【评论列表】Weiba/comment_list

接口名称：	Weiba/comment_list	
接口描述：	获取帖子的评论列表	
演示地址：		
输入参数：	(int) id , 帖子 ID (int) count , 分页显示时 , 指定每页显示条数 (默认 20)	

	(int) page , 分页显示时 , 指定获取的页码 (默认取第 1 页) (string) format , 返回格式可选 json/php/test 格式 (默认 json 格式)	
输出结果 :		
字段说明 :	成功时 :	
	失败时 :	
其他说明 :		

5.7.8 【评论帖子】Weiba/create_comment

接口名称 :	Weiba/create_comment	
接口描述 :	评论帖子操作	
演示地址 :		
输入参数 :	(int) id , 帖子 ID (string) user_id , 用户 ID (选填 , 默认当前用户 ID) (string)content , 评论内容	
输出结果 :		
字段说明 :	成功时 :	1
	失败时 :	0
其他说明 :		

5.7.9 【回复评论】Weiba/reply_comment

接口名称 :	Weiba/reply_comment
--------	---------------------

接口描述：	评论帖子操作	
演示地址：		
输入参数：	(int) id , 帖子 ID (string) user_id , 用户 ID (选填 , 默认当前用户 ID) (string)content , 评论内容	
输出结果：		
字段说明：	成功时：	1
	失败时：	0
其他说明：		

5.7.10 【删除评论】Weiba/delete_comment

接口名称：	Weiba/delete_comment	
接口描述：	删除评论	
演示地址：		
输入参数：	(int) id , 评论 ID (string) user_id , 用户 ID (选填 , 默认当前用户 ID)	
输出结果：		
字段说明：	成功时：	1
	失败时：	0
其他说明：		

5.7.11 【关注微吧的帖子】Weiba/following_posts

接口名称：	Weiba/following_posts	
接口描述：	获取关注微吧的帖子列表	
演示地址：		
输入参数：	(string) user_id , 用户 ID (选填 , 默认当前用户 ID) (int) count , 分页显示时 , 指定每页显示条数 (默认 20) (int) page , 分页显示时 , 指定获取的页码 (默认取第 1 页) (string) format , 返回格式可选 json/php/test 格式 (默认 json 格式)	
输出结果：		
字段说明：	成功时：	
	失败时：	
其他说明：		

5.7.12 【发布的帖子】Weiba/ posts

接口名称：	Weiba/posts	
接口描述：	获取发布的帖子列表	
演示地址：		
输入参数：	(string) user_id , 用户 ID (选填 , 默认当前用户 ID) (int) count , 分页显示时 , 指定每页显示条数 (默认 20) (int) page , 分页显示时 , 指定获取的页码 (默认取第 1 页)	

	(string) format , 返回格式可选 json/php/test 格式 (默认 json 格式)	
输出结果 :		
字段说明 :	成功时 :	
	失败时 :	
其他说明 :		

5.7.13 【评论的帖子】Weiba/commenteds

接口名称 :	Weiba/commenteds	
接口描述 :	获取评论的帖子列表	
演示地址 :		
输入参数 :	(string) user_id , 用户 ID (选填 , 默认当前用户 ID) (int) count , 分页显示时 , 指定每页显示条数 (默认 20) (int) page , 分页显示时 , 指定获取的页码 (默认取第 1 页) (string) format , 返回格式可选 json/php/test 格式 (默认 json 格式)	
输出结果 :		
字段说明 :	成功时 :	
	失败时 :	
其他说明 :		

5.7.14 【搜索微吧】Weiba/search_weiba

接口名称 :	Weiba/search_weiba
--------	---------------------------

接口描述：	搜索微吧	
演示地址：		
输入参数：	(string) user_id , 用户 ID (选填 , 默认当前用户 ID) (string) keyword , 搜索关键字 (int) count , 分页显示时 , 指定每页显示条数 (默认 20) (int) page , 分页显示时 , 指定获取的页码 (默认取第 1 页) (string) format , 返回格式可选 json/php/test 格式 (默认 json 格式)	
输出结果：		
字段说明：	成功时：	
	失败时：	
其他说明：		

5.7.15 【搜索帖子】Weiba/search_post

接口名称：	Weiba/ search_post	
接口描述：	搜索帖子	
演示地址：		
输入参数：	(string) user_id , 用户 ID (选填 , 默认当前用户 ID) (string) keyword , 搜索关键字 (int) count , 分页显示时 , 指定每页显示条数 (默认 20) (int) page , 分页显示时 , 指定获取的页码 (默认取第 1 页)	

	(string) format , 返回格式可选 json/php/test 格式 (默认 json 格式)	
输出结果 :		
字段说明 :	成功时 :	
	失败时 :	
其他说明 :		

5.7.16 【收藏帖子】Weiba/post_favorite

接口名称 :	Weiba/post_favorite	
接口描述 :	收藏帖子	
演示地址 :		
输入参数 :	(string) user_id , 用户 ID (选填 , 默认当前用户 ID) (int) id, 帖子 ID (string) format , 返回格式可选 json/php/test 格式 (默认 json 格式)	
输出结果 :		
字段说明 :	成功时 :	1
	失败时 :	0
其他说明 :		

5.7.17 【取消收藏】Weiba/post_unfavorite

接口名称 :	Weiba/post_unfavorite	
接口描述 :	取消收藏的帖子	

演示地址：		
输入参数：	(string) user_id , 用户 ID (选填 , 默认当前用户 ID) (int) id , 帖子 ID (string) format , 返回格式可选 json/php/test 格式 (默认 json 格式)	
输出结果：		
字段说明：	成功时：	1
	失败时：	0
其他说明：		

5.7.18 【收藏帖子列表】Weiba/favorite_list

接口名称：	Weiba/favorite_list	
接口描述：	收藏帖子列表	
演示地址：		
输入参数：	(string) user_id , 用户 ID (选填 , 默认当前用户 ID) (string) format , 返回格式可选 json/php/test 格式 (默认 json 格式)	
输出结果：		
字段说明：	成功时：	
	失败时：	
其他说明：		

5.8 签到

5.8.1 【签到信息】Checkin/get_check_info

接口名称：	Weiba/get_posts	
接口描述：	获取签到信息	
接口编号：		
演示地址：		
输入参数：	<p>(int) mid , 用户 ID (选填 , 默认当前用户 ID)</p> <p>(string) type , 包括待解决、已解决、精华知识</p> <p>(int) since_id , 起始问题 ID</p> <p>(int) max_id , 最大问题 ID</p> <p>(int) count , 分页显示时 , 指定每页显示条数 (默认 20)</p> <p>(int) page , 分页显示时 , 指定获取的页码 (默认取第 1 页)</p> <p>(string) format , 返回格式可选 json/php/test 格式 (默认 json 格式)</p>	
输出结果：		
字段说明：	成功时：	
	失败时：	
其他说明：		

5.8.2 【签到操作】Checkin/checkin

接口名称：	Weiba/get_posts	
接口描述：	获取签到信息	
接口编号：		
演示地址：		
输入参数：	<p>(int) mid , 用户 ID (选填 , 默认当前用户 ID)</p> <p>(string) type , 包括待解决、已解决、精华知识</p> <p>(int) since_id , 起始问题 ID</p> <p>(int) max_id , 最大问题 ID</p> <p>(int) count , 分页显示时 , 指定每页显示条数 (默认 20)</p> <p>(int) page , 分页显示时 , 指定获取的页码 (默认取第 1 页)</p> <p>(string) format , 返回格式可选 json/php/test 格式 (默认 json 格式)</p>	
输出结果：		
字段说明：	成功时：	
	失败时：	
其他说明：		