这里是很简陋的看知乎官方API文档0.1.5版。

当前版本不需要身份认证即可调用,但请大家注意节制,不要滥用。

最新文章 用户动态 更新日志 常见问题

本文档适合以下人群阅读:

- 对「看知乎」的网站感兴趣,准备开发APP或客户端的开发者;
- 对知乎用户数据和排行感兴趣的数据分析者; ■ 不知道干什么就是想随便看看的闲人。

基本用法

API域名: http://api.kanzhihu.com

调用格式:目前只支持GET,URL格式为 http://api.kanzhihu.com/API名称,或者 http://api.kanzhihu.com/API名称/参数 返回数据: json

API文档

留言板

错误: {error: "..."}

正确: {error: "", data: ... }

服务器返回值之后首先检查error字段,如果不为空说明出错,为空再处理返回数据。

API列表

getposts

用途: 获取「看知乎」首页文章列表,每次取10篇。

参数:时间戳(可选,留空时取最新10篇,有值时则取此时间戳之前的10篇)

返回值: ■ count(number),本次获取文章数量(一般为10,也可能小于10)

■ posts(array),文章信息列表,字段如下:

- date(string),发表日期(yyyy-mm-dd) ■ name(string), 文章名称 (yesterday, recent, archive)
- pic(string), 抬头图url
- publishtime(number),发表时间戳 ■ count(number),文章包含答案数量
- excerpt(string), 摘要文字 如此时间戳之前无文章,则error=="no result"
- 描述:「看知乎」的答案推荐以文章为单位,每天在三个时段发布三篇,名字分别为<u>昨日最新</u>(yesterday)、<u>近日热门</u>(recent)和
- <u>历史精华</u>(archive),每篇推荐32~40个答案不等。本API就用于获取文章列表。如果要取更早的10篇(即向前翻页),则把当前最早 一篇的时间戳作为参数即可。

示例:

http://api.kanzhihu.com/getposts http://api.kanzhihu.com/getposts/1411261200

checknew

参数: 时间戳(必选)

返回值:

用途: 检查「看知乎」首页在指定时间之后有没有更新

章,如果返回结果为true则再次调用getposts。 建议在每天稍晚于5:00、11:00、17:00的时间分别进行查询,或者为了考虑日后的兼容性,每小时查询一次。

示例: http://api.kanzhihu.com/checknew/1444899600

用途: 获取单篇文章的答案列表

■ result(bool),是否有更新

getpostanswers

描述:每次调用getposts获取最新文章之后,客户端应保存本次最新一篇发表的时间戳,之后通过调用此API查询是否已经发布了新文

参数:日期(必选,8位数字,如20140925)、名称(必选:yesterday、recent、archive)

返回值:

■ answers(array), 答案列表, 字段如下: ■ title(string),文章id

■ time(datetime),发表时间 ■ summary(string), 答案摘要

■ count(number), 答案数量

questionid(string),问题id,8位数字 ■ answerid(string), 答案id, 8位数字

■ authorname(string), 答主名称

- authorhash(string), 答主hash
- avatar(string), 答主头像url ■ vote(number), 赞同票数

如id不存在或没有答案,则error=="no result"

- 描述:通过文章列表获得某篇具体文章的日期及名称,然后通过本API获取文章中的答案列表。答案并未排序或者按照问题合并,开发 者可以自行处理。
 - 问题url: http://www.zhihu.com/question/questionid 答案url: http://www.zhihu.com/question/<u>questionid</u>/answer/<u>answerid</u>

根据返回数据可以拼接以下url:

示例: http://api.kanzhihu.com/getpostanswers/20150925/archive

userdetail2 用途: 获取用户详细页信息

参数:用户hash(必选)

■ name(string): 用户名 ■ avatar(string): 用户头像url

用户主页url: http://www.zhihu.com/people/authorhash

■ description(string): 用户个人描述 ■ detail(object): 用户详细数据 ■ ask(number): 提问数

返回值:

■ answer(number): 回答数 ■ post(number): 专栏文章数

■ signature(string): 用户签名

- agree(number): 赞同数 ■ agreei(number): 1日赞同数增加 ■ agreeiratio(string): 1日赞同数增幅
- ratio(number): 平均赞同(总赞同数/(回答+专栏)) ■ followee(number): 关注数 ■ follower(number): 被关注数(粉丝)
- followiratio(string): 1日被关注数增幅 ■ followeriw(number): 7日被关注数增加 ■ followiratiow(string): 7日被关注数增幅

■ agreeiw(number): 7日赞同数增加

■ agreeiratiow(string): 7日赞同数增幅

■ followeri(number): 1日被关注数增加

- thanks(number): 感谢数 ■ tratio(number): 感谢/赞同比 ■ fav(number): 收藏数
- fratio(number): 收藏/赞同比 ■ logs(number): 公共编辑数 ■ mostvote(number): 最高赞同
- mostvote5(number): 前5赞同 ■ mostvote5percent(string): 前5赞同占比 ■ mostvote10(number): 前10赞同

■ mostvotepercent(string): 最高赞同占比

■ mostvote10percent(string): 前10赞同占比

count10000(number): 赞同≥10000答案数 count5000(number): 赞同≥5000答案数 ■ count2000(number): 赞同≥2000答案数

■ count1000(number): 赞同≥1000答案数

count500(number): 赞同≥500答案数

count200(number): 赞同≥200答案数

count100(number): 赞同≥100答案数 ■ star(object): 七星阵排名 ■ answerrank: 回答数+专栏文章数排名

■ agreerank: 赞同数排名

■ ratiorank: 平均赞同排名

■ followerrank: 被关注数排名

■ trend(array), 近日动态,字段如下:

■ favrank: 收藏数排名 ■ count1000rank: 赞同超1000的回答数排名 ■ count100rank: 赞同超100的回答数排名

■ date(string), 日期, 格式为2014-09-25

■ answer(number): 回答数+专栏文章数

■ date(string): 日期,格式为2014-09-25

用途: 获取某项指标排名前列的用户列表(分页)

■ topuser(array),用户列表,字段如下:

■ id(string): 用户ID

■ name(string): 用户名称

hash(string): 用户hash

如页数为空,则默认取第一页;

如每页条数为空,则默认每页30条;

■ avatar(string): 用户头像url

■ signature(string): 用户签名

如传递agree,则这里的字段名就是agree。

所有指标类型列表与userdetail2的返回值里detail属性内容一致。

取赞同排名前30的用户http://api.kanzhihu.com/topuser/agree/1

取关注排名161~200的用户http://api.kanzhihu.com/topuser/follower/5/40

如用户hash不存在,则error=="no result"

■ topanswers(array), 高票答案,字段如下: ■ title(string),标题 ■ link(string): 链接地址(不含域名)

■ follower(number): 被关注数

■ agree(number): 赞同数

■ agree(number): 赞同数

- 描述:通过答案列表或其他途径获得知乎用户hash,然后通过本API获取用户详细信息用于展示及分析。 其中detail包含了用户在最近一条快照中的各项数据; star选择了detail中的七项代表性数据进行排名(由于看知乎的数据库并没有包含知乎全站,所以排名越靠后越不准确),之所以叫七
- topanswers是最高票的10个答案(或专栏文章)。 示例:

topuser

返回值: ■ count(number), 用户数量

■ value(number): 某项指标的值,注意,这里的字段名不是"value",而是调用API时传递的参数"<u>指标类型</u>";

参数: 指标类型、页数(可选,最小为1)、每页条数(可选,最小为1,最大为50)

■ ispost(bool): 是否专栏文章,用于判断链接域名是zhihu.com还是zhuanlan.zhihu.com

如用户存在,但当日没有他的数据(可能是数据抓取出错),则error=="no snapshot"

星阵,是因为在最初版本的设计中这七项以雷达图形式显示,外观很像星星;

trend显示了用户最多最近30天内的动态,可以看出回答、赞同、粉丝的增长情况;

http://api.kanzhihu.com/userdetail2/1f644a1b7da169d2b56e1a4c6da61fea

- 如所选范围超出了前500名,则出错并返回空值; 如所选范围刚好落在500名前后,可能返回数量会小于每页条数;
- 描述:本API实际是对用户分析所做的按列筛选和分页处理,适用于移动客户端。如果为了完整分析数据,仍然推荐直接使用用户分析 或通过其中的json数据文件进行。 示例:

用途: 搜索用户 参数: 搜索关键字

■ name(string): 用户名称 hash(string): 用户hash avatar(string): 用户头像url

返回值:

searchuser

- signature(string): 用户签名 ■ answer(number): 回答+专栏数量
- agree(number): 赞同数量 ■ follower(number): 被关注数量

■ count(number), 用户数量

■ id(string): 用户ID

■ users(array),用户列表,字段如下:

- 如搜索不到结果,则error="no user"
- 搜索结果按如下顺序排列: 首先是用户ID等于关键字的; 然后是用户名等于关键字的,按被关注数排序;
- 再后面是用户名包含关键字的,按被关注数排序。 示例:
- http://api.kanzhihu.com/searchuser/aton http://api.kanzhihu.com/searchuser/苏莉安 http://api.kanzhihu.com/searchuser/苏

本站数据来源于 知乎网, 版权归原作者所有

Powered by WordPress. Theme by Alx.