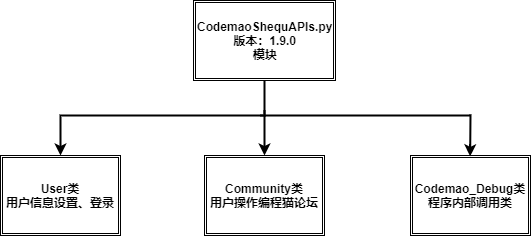
**Python For Codemao shequ APIs 使用方法**

**版本:1.9.0**

1. 安装方法

请运行该目录下的 “Setup.py”进行安装，并且确保所在目录下有“CodemaoShequAPIs.py”文件！若出现无法安装的情况，请尝试以管理员身份运行！

1. 结构介绍



User类：控制编程猫社区的用户版块，设计到登录、获取用户数据、更改用户数据。

Community类：控制编程猫社区的论坛类，用于发帖、删帖、发评论、删评论、获取文章信息、获取版块信息。

Codemao\_Debug类：程序内部的调试，用户不可调用，不多做赘述。

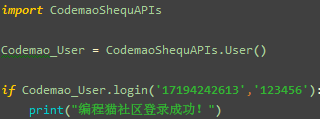
1. User类中的方法介绍
   1. login(self,Username,Password) 方法

通过账号密码，获取用户的cookie，cookie作为用户登录到服务器的标识。  
获得cookie后在Codemao类销毁前所有子类均记录cookie。即登录用户未操作论坛做铺垫。

参数说明：User(str) --- 提供用户名Possword(str) --- 提供用户密码

返回值说明：bool类型 -- 是否登录成功。

示例：



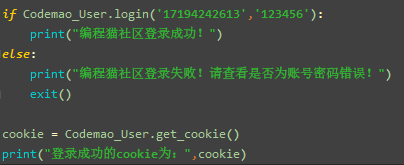
* 1. get\_cookie(self) 方法

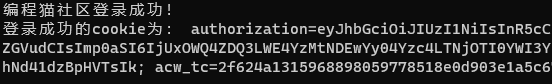
返回已设置或登录后的cookie，可以获取字符串形式的cookie可用于保存使用。

参数说明：-无参数-

返回值说明：str类型 – 程序正在使用的cookie

示例：





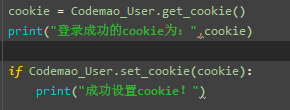
* 1. set\_cookie(self,cookie) 方法

设置程序使用的cookie ，可以直接将浏览器的cookie直接复制到此，可以达到cookie登录的效果！

参数说明：cookie(str) – 传入的cookie

返回值说明：bool – 是否设置成功

实例：



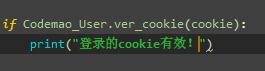
* 1. ver\_cookie(self, cookie=web.cookies) 方法

验证cookie是否有效，用于测试用户登录是否过期。

参数说明：cookie(str) – \*可选参数\* 若不提供cookie则采用默认cookie

返回值说明：bool – cookie是否有效

实例：



* 1. get\_my\_info(self) 方法

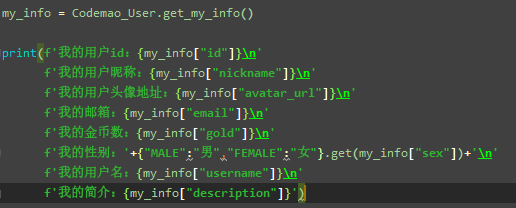
获取登录用户的个人信息，如果用户未登录则会返回一个空的字典，或者报错。

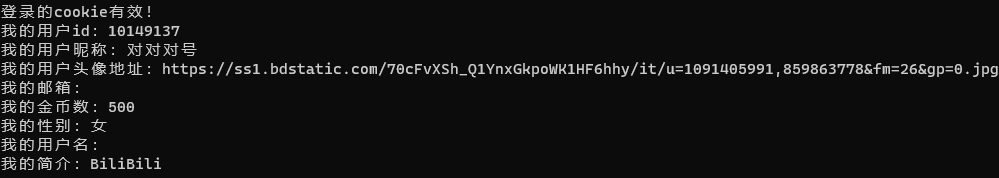
部分用户信息未设置则没有

参数说明：\*无参数\*

返回值说明：dict – 复杂用户数据 [参见数据类型第①条](#类型①)

实例：





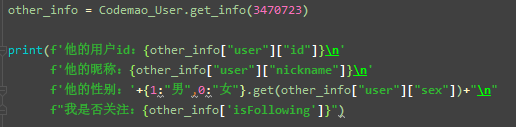
* 1. get\_info(self, uid) 方法

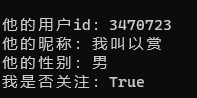
获取某一个用户的信息，不是全部是公开信息。相比自己的信息较少。

参数说明：uid(int) -- 用户id

返回值说明：dict – 返回一个用户信息的字典 [参见数据类型②](#类型②)

实例：





* 1. set\_my\_info(self, nickname, sex, description, fullname, birthday, avatar\_url) 方法

设置个人信息

参数说明：

nickname(str) -- 昵称

sex(int) -- 性别(0女 1男)

description(str) -- 描述

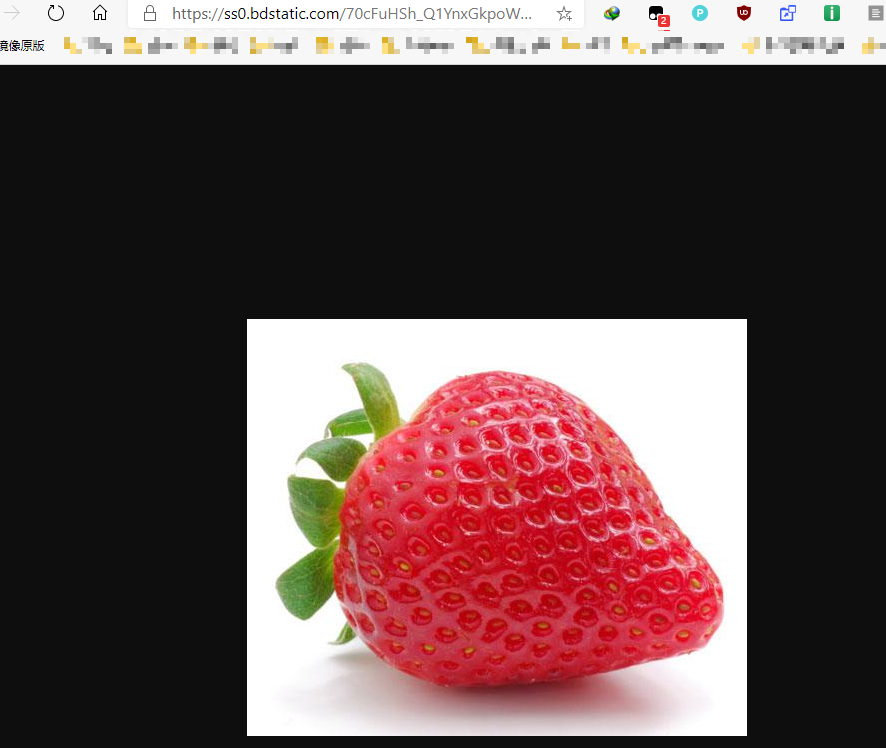
fullname(str) -- 真实名称

birthday(int) -- 生日(时间戳)

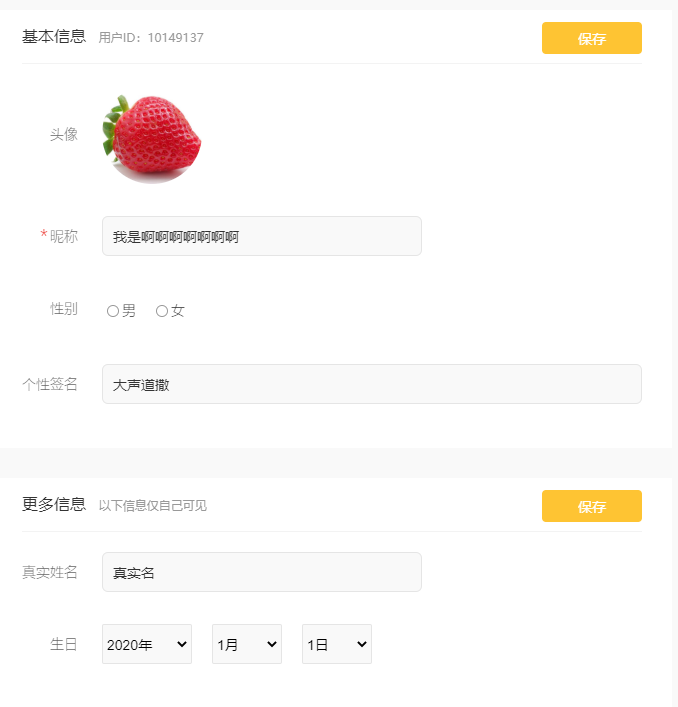
vatar\_url(str) -- 头像网址

返回值：bool – 是否成功

实例：







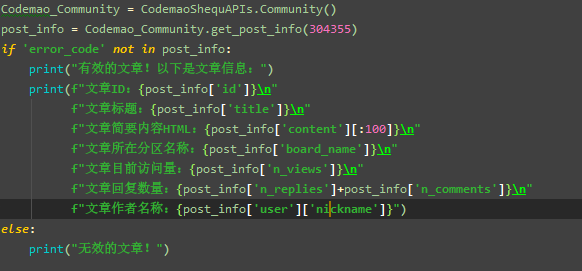
1. Community类中的方法介绍
   1. get\_post\_info(self, post\_id) 方法

获取文章标准信息，获取时会增加文章访问量。注意：此接口调用的是浏览文章时获取信息的API，是获得较为详细的文章信息，并不是获取首页的文章摘要信息。

参数说明：post\_id(int) -- 帖子的ID

返回:帖子信息(dict) -- 返回帖子字典 [详情见数据类型③](#类型③)

实例：





* 1. get\_post\_comment(self, post\_id) 方法

此方法用于获取文章的全部评论，如果评论太多则获取的时间会稍微久一点。

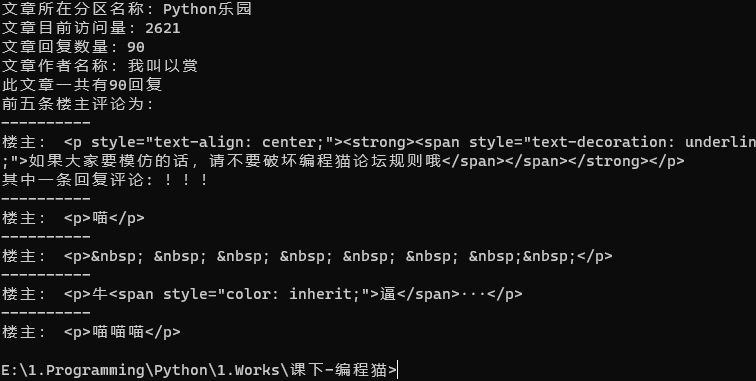
参数说明：post\_id(int) -- 帖子的ID

返回值说明：评论(list) -- 评论,每一个项中都存储了一个评论数据(dict)，

评论数据类型见 [数据类型④](#类型④)

示例：



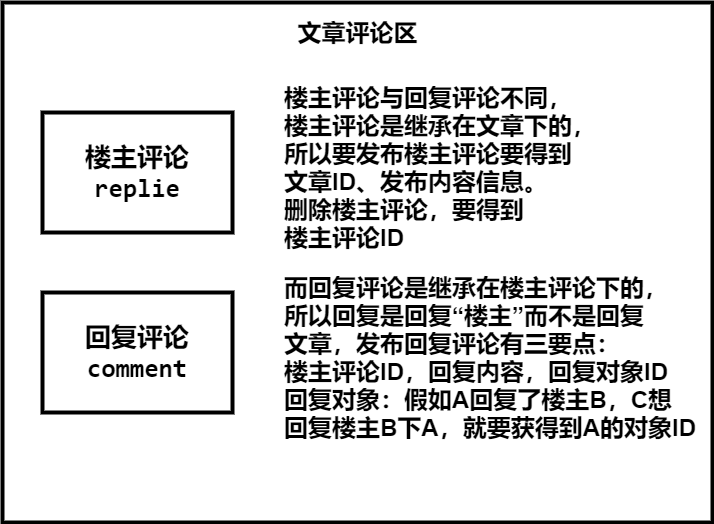




(小插曲) replie与comment（楼主评论和回复评论）

如果你理解了这一步分内容，你将更好的调用下面的API。

如图：





***而且replie与comment都有对应的API，两者不可交换使用！！！***

* 1. replie(self, post\_id, html) 方法

用于回复文章，即楼主评论

参数说明：post\_id(int) – 文章ID ； html(str) – 评论内容(HTML)

返回值说明：int – 成功返回楼主评论ID

实例：



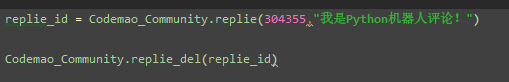
* 1. replie\_del(self, replies\_id) 方法

用于删除文章评论，即删除评论评论

参数说明：post\_id(int) – 文章ID

返回值：是否成功(bool)

实例：



* 1. comment(self, replies\_id, html, comments\_id=0) 方法

用户回复评论，即回复楼主评论；或是用于回复已经回复楼主的其它人评论。

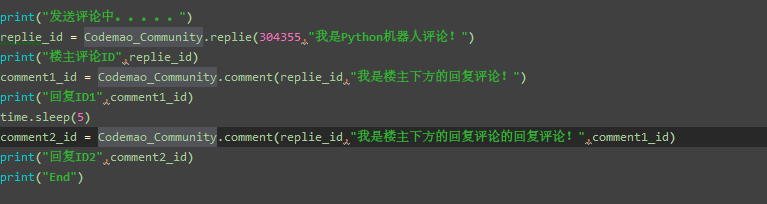
replies\_id(int) -- 楼主评论ID

comments\_id(int) -- 可选 如果楼主下面还有人回复，你想要回复那个人，请提供他的id，没有默认为 0。如果要回复别人请不要回复的太快，否则会失败！

html(str) -- 回复的内容 请用html代码

返回：评论id(int) -- 成功返回评论ID，失败返回0

实例：





* 1. comment\_del(self, comments\_id) 方法

用于删除回复评论

参数说明：comments\_id(int) –回复评论

返回：是否成功(bool)

* 1. replies\_like(self, replies\_id) 方法

为楼主评论点赞

参数说明：replies\_id(int) -- 楼主评论id

返回：是否成功(bool)

* 1. replies\_diss\_like(self, replies\_id) 方法

为楼主评论取消点赞

参数说明：replies\_id(int) -- 楼主评论id

返回：是否成功(bool)

* 1. comment \_like(self, replies\_id) 方法

为回复评论点赞

参数说明：replies\_id(int) -- 楼主评论id

返回：是否成功(bool)

* 1. comment \_diss\_like(self, replies\_id) 方法

为回复评论取消点赞

参数说明：replies\_id(int) -- 楼主评论id

返回：是否成功(bool)

* 1. get\_my\_replies\_post(self) 方法

获取我评论过的文章

返回：返回我评论的帖子的数据(list) -- 帖子数据 [详细见类型⑤](#类型⑤)

* 1. get\_my\_release\_post(self) 方法

获取我评论过的文章

返回：返回我评论的帖子的数据(list) -- 帖子数据 [详细见类型⑤](#类型⑤)

* 1. get\_latest\_replies\_post(self) 方法

获取首页最新文章，包括指定一共300个。

返回：列表(list) -- 每一个项都是文章id

* 1. get\_post\_easy\_info(self, post\_ids) 方法

获取文章标准信息，

匹配数量不要 > 30

参数说明：

post\_ids -- 允许正则表达式，也允许传入列表，列表内包含文章id

返回：文章信息(list) -- 文章信息(list) -- 批量文章信息，每一个项都是文章标准信息 [详细见类型⑤](#类型⑤)

* 1. get\_7dayHot\_post(self, total, board\_id="") 方法

获取七日热门文章 返回文章信息，不是文章id

参数说明：

total -- 获取数量

board\_id -- 可选参数，用于获取某版块7日热门，不指定请留空

[版块ID见类型⑥](#类型⑥)

返回：文章信息(list) -- 文章信息,每一个项都是文章标准信息 [详细见类型⑤](#类型⑤)

* 1. get\_7datHot\_many(self, board\_id="") 方法

获取七日热门文章总数

board\_id -- 可选参数，用于获取某版块7日热门，不指定请留空

[版块ID见类型⑥](#类型⑥)

返回：总数(int) -- 文章总数

* 1. get\_selections\_post(self, total, board\_id="") 方法

获取精品合集文章 返回文章信息，不是文章id

参数说明：

total -- 获取数量

board\_id -- 可选参数，不指定请留空

[版块ID见类型⑥](#类型⑥)

返回：文章信息(list) -- 文章信息,每一个项都是文章标准信息 [详细见类型⑤](#类型⑤)

* 1. get\_selections \_many(self, board\_id="") 方法

获取精品合集文章总数

board\_id -- 可选参数，不指定请留空

[版块ID见类型⑥](#类型⑥)

返回：总数(int) -- 文章总数

* 1. get\_ask\_help\_post(self, total, board\_id="") 方法

获取求帮助文章 返回文章信息，不是文章id

参数说明：

total -- 获取数量

board\_id -- 可选参数，不指定请留空

[版块ID见类型⑥](#类型⑥)

返回：文章信息(list) -- 文章信息,每一个项都是文章标准信息 [详细见类型⑤](#类型⑤)

* 1. get\_ask\_help\_many(self, board\_id="") 方法

获取求帮助文章总数

board\_id -- 可选参数，不指定请留空

[版块ID见类型⑥](#类型⑥)

返回：总数(int) -- 文章总数

* 1. search\_post (self, key,total) 方法

获取求帮助文章 返回文章信息，不是文章id

参数说明：

key(str) -- 搜索内容

total(int) -- 搜索数量

返回：文章信息(list) -- 文章信息,每一个项都是文章标准信息 [详细见类型⑤](#类型⑤)

* 1. search\_post\_many(self, key) 方法

获取求帮助文章总数

参数说明：

key(str) -- 搜索内容

返回：总数(int) -- 文章总数

* 1. get\_board\_info (self, board\_id) 方法

获取版块信息

board\_id – 版块ID [版块ID见类型⑥](#类型⑥)

返回：版块信息(dict) – 版块信息 [见类型⑦](#类型⑦)

* 1. get\_board\_post\_info(self, board\_id, total) 方法

获取版块文章

board\_id – 版块ID [版块ID见类型⑥](#类型⑥)

total – 获取总量

返回：文章信息(list) -- 文章信息,每一个项都是文章标准信息 [详细见类型⑤](#类型⑤)

* 1. reports\_post(self, post\_id, reports\_id, reports\_description) 方法

举报文章，不是举报评论

post\_id – 文章ID

reports\_id – 举报版块 [见类型⑧](#类型⑧)

reports\_description – 举报描述

返回：举报ID

* 1. reports\_discussions(self,discussions\_id,reports\_id,reports\_description="") 方法

举报评论

discussions\_id(int) -- 评论ID

reports\_id(int) -- 举报分区，如下：

reports\_description -- 举报描述 reports\_id选择0时必须填写 选择其它不填写

返回：举报ID

* 1. post(self, board\_id, post\_title, post\_html) 方法

发布文章

参数说明：

board\_id(int) -- 版块ID [版块ID见类型⑥](#类型⑥)

post\_title(str) -- 文章标题 5~50字

post\_html(str) -- 文章内容(html)

发布成功返回文章id，失败报错

* 1. post\_del(self, post\_id) 方法

删除文章

参数说明

post\_id – 文章ID

返回：是否成功(bool)

1. 数据类型
   1. 复杂用户数据
      * 用户未设置则返回空文本
      * 数据key -- 数据分类 -- 数据类型
      * id -- 用户ID -- int
      * nickname -- 用户昵称 -- str
      * avatar\_url -- 用户头像URL -- str
      * email -- 用户邮箱 -- str
      * gold -- 用户金币数 -- int
      * qq -- 用户QQ -- str
      * real\_name -- 用户真名 -- str
      * sex -- 用户性别 -- str(男MALE 女FEMALE)
      * username -- 用户名 -- str
      * voice\_forbidden -- 已被禁言 -- bool
      * birthday -- 生日 -- int (时间戳)
      * description -- 简介 -- str
      * phone\_number -- 加掩码的电话 -- str
      * create\_time -- 用户创建时间 -- int(时间戳)
      * oauths -- 用户账户绑定情况 -- list(列表内有数组见下)
        + | 原数据：[ {'id':1,'name':'wechat','is\_bound':False} , ...]
        + | 翻译： id -- 1:微信 2:QQ 3:优学派
        + | 4:国家教育资源公共服务平台 5:豹豹龙官网
        + | 6:新华文轩出版传媒 7:云知声
        + | 8.icollege远程教育平台 9.dledc? 10.apple?
        + | 11.我time网 12.中国招商银行 13:编程猫-新零售
        + | name -- 上述对应的名称 is\_bound -- 是否绑定
      * has\_password -- 用户有密码 -- bool
      * user\_type -- 用户类型 -- int (未知，盲猜有普通用户、管理员之分)
      * show\_guide\_flag -- 展示向导标志 -- int (未知，盲猜引导页面)
      * doing -- “我正在做什么”列表 -- str
      * level -- 用户等级 -- int
      * telephone -- 绑定电话 -- str
   2. 公开用户数据

* isFollowing -- 我是否已关注 -- int
* collectionTimes -- 收藏数 -- int
* forkedTimes -- 再创作数 -- int
* praiseTimes -- 获赞数 -- int
* viewTimes -- 被浏览次数 -- int
* user -- 用户信息 -- dict 如下：
* | id -- 用户id ； nickname -- 用户昵称 ； sex -- 性别(int 0女 1男) ； description -- 描述 ；doing -- 正在做的 ;
* | avatar -- 头像地址 ； level -- 等级 ；preview\_work\_id -- 展示作品id ； preview\_work\_src -- 展示作品地址
* work -- 展示作品详情 -- dict 如下：
* | id -- 作品id name -- 作品名称 preview -- 作品预览图
  1. 详细文章信息
* 帖子不存在时：
* 数据key -- 数据分类 -- 数据类型
* error\_code -- 错误代码 -- str
* error\_message -- 错误信息 -- str
* log\_uuid -- 用户登录uuid -- str
* domain -- ??? -- list
* 帖子存在时：
* 数据key -- 数据分类 -- 数据类型
* id -- 帖子id -- int
* title -- 帖子标题 -- str
* content -- 帖子内容(HTML) -- str
* board\_id -- 分区id -- int
* board\_name -- 分区名称 -- str
* updated\_at -- 更新时间 -- int(时间戳)
* created\_at -- 创建时间 -- int(时间戳)
* n\_views -- 帖子访问量 -- int
* n\_replies -- 评论回复数 -- int
* n\_comments -- 评论数(楼主) -- int
* is\_authorized -- 被授权的文章 -- bool
* is\_featured -- 有特色的文章? -- bool
* is\_hotted -- 热门的文章 -- bool
* is\_pinned -- 置顶的文章 -- bool
* tutorial\_flag -- 被辅导标志 -- int
* ask\_help\_flag -- 求助标志 -- int
* user -- 帖子用户信息 -- dict
* | id -- 用户id ； nickname -- 用户名称 ；
* | avatar\_url -- 用户头像 ； subject\_id -- 课程id ；
* | work\_shop\_name -- 参加工作室的名称 ；
* | work\_shop\_level -- 工作室等级 ; wuhan\_medal -- 拥有武汉奖章
  1. 标准评论数据类型
* 数据key -- 数据分类 -- 数据类型
* id -- 用户id -- int
* is\_top -- 评论被置顶 -- bool
* n\_likes -- 点赞数 -- int
* is\_liked -- 我喜欢 -- bool
* content -- 评论内容(html) -- str
* n\_comments -- 回复评论数 -- int
* updated\_at -- 更新时间 -- int(时间戳)
* created\_at -- 创建时间 -- int(时间戳)
* earliest\_comments -- 回复的评论 -- dict(与上述一致)
  1. 标准文章信息
* 数据key -- 数据分类 -- 数据类型
* id -- 帖子id -- int
* title -- 帖子标题 -- str
* content -- 帖子摘要(HTML) -- str
* created\_at -- 创建时间 -- int(时间戳)
* n\_replies -- 评论回复数 -- int
* n\_comments -- 评论数(楼主) -- int
* is\_authorized -- 被授权的文章 -- bool
* is\_featured -- 有特色的文章? -- bool
* is\_hotted -- 热门的文章 -- bool
* is\_pinned -- 置顶的文章 -- bool
* tutorial\_flag -- 被辅导标志 -- int
* ask\_help\_flag -- 求助标志 -- int
* user -- 帖子用户信息 -- dict
* | id -- 用户id ； nickname -- 用户名称 ；
* | avatar\_url -- 用户头像 ； subject\_id -- 课程id ；
* | work\_shop\_name -- 参加工作室的名称 ；
* | work\_shop\_level -- 工作室等级 ; wuhan\_medal -- 拥有武汉奖章
* commented\_at -- 评论时间 -- int(时间戳)
* commented\_user -- 评论用户信息 -- dict 如下：
* | id -- 评论用户id ； nickname -- 用户昵称 ；
* | avatar\_url -- 头像地址 ； subject\_id -- 课程id ；
* | work\_shop\_name -- 参加工作室的名称 ；
* | work\_shop\_level -- 工作室等级 ; wuhan\_medal -- 拥有武汉奖章
* replied\_at -- 最新回复（非自己回复） -- int(时间戳)
* | id -- 评论用户id ； nickname -- 用户昵称 ；
* | avatar\_url -- 头像地址 ； subject\_id -- 课程id ；
* | work\_shop\_name -- 参加工作室的名称 ；
* | work\_shop\_level -- 工作室等级 ; wuhan\_medal -- 拥有武汉奖章
  1. 版块ID

1 -- 点猫看板 ； 2 -- 积木编程乐园 ； 3 -- 代码岛 ； 4 -- 通天塔 ；

5 -- 你问我答 ； 6 -- 动漫小说 ；

7 -- 灌水池塘 ； 8 -- 我要提建议 ； 9 -- 作品秀 ；10 -- 工作室 ；

11 -- Python乐园 ； 12 -- 求素材；

13 -- NOC编程猫比赛 ； 15 -- UnderTale ； 16 -- I wanna ；

17 -- 热门活动 ； 18 -- Scratch专区 ；

19 -- 动作游戏专区 ； 20 -- 射击游戏专区 ； 21 -- 休闲游戏专区 ；

22 -- 学术作品专区 ；23 -- 艺术作品专区 ；

24 -- 工作室招新； 25 -- 师徒广场 ； 26 -- 源码精灵

* 1. 版块信息
* 不存在版块：
* 数据key -- 数据分类 -- 数据类型
* error\_code -- 错误代码 -- str
* error\_message -- 错误信息 -- str
* log\_uuid -- 用户登录uuid -- str
* 存在版块：
* 数据key -- 数据说明 -- 数据类型
* id -- 版块ID -- int
* name -- 版块名称 -- str
* description -- 版块简介 -- str
* icon\_url -- 版块图片网址 -- str
* is\_hot -- 版块是否热门 -- bool
* has\_popular -- 版块是否流行 -- bool
* has\_selection -- 版块是否被选拔? -- bool
* n\_posts -- 版块帖子数 -- int
* n\_discussions -- 版块讨论数 – int
  1. 举报信息

1 -- 违法违规 ； 2 -- 色情暴力 ； 3 -- 侵犯隐私 ； 4 -- 人身攻击 ；

5 -- 引战 ； 6 -- 垃圾广告 ； 7 -- 无意义刷屏 ； 8 -- 不良信息 ；

0 -- 自定义（举报评论时特有，须填写内容）