FLOW

应用接口定义

版 本 历 史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 提交日期 | 版本号 | 作 者 | 修 改 内 容 | 备注 |
| 2017-7-6 | 1.0 |  | 初始版本 |  |
|  |  |  |  |  |

术语缩写与解释

|  |  |  |  |
| --- | --- | --- | --- |
| 缩 写 | 解 释 | 缩 写 | 解 释 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目 录

1) 认证接口(Auth) 6

1.1 注册验证码 6

1.2 账号注册 6

1.3 认证登录 7

~~1.4~~ ~~忘记密码~~ 7

1.5 重置密码 8

2) 基础模块(FND) 9

2.1 获取用户列表 9

2.2 获取用户信息 10

2.3 修改用户信息 10

2.4 标签分组 11

2.5 设置用户标签 12

2.6 文件上传 12

2.7 职责类型 13

2.8 未读通知数量 13

2.9 获取通知列表 14

2.10 更新通知已读 15

3) 互动模块(POFO) 16

3.1 发帖 16

3.2 关注 / 取消关注 16

3.3 列表 17

3.4 详情 18

3.5 点赞 / 取消点赞 19

3.6 发表评论 20

3.7 评论列表 20

~~3.8~~ ~~评论追踪~~ 22

4) 问答模块(SOLV) 24

4.1 发布问题 24

4.2 标记SPAM 24

4.3 答复 25

4.4 评论 26

4.5 评论列表 26

4.6 问题列表 27

4.7 问题详情 28

4.8 接受答复 30

4.9 答复vote 30

5) 私信(PRVM) 32

5.1 发私信 32

5.2 私信数量 32

5.3 私信列表 33

# 统一约定

## 密码变换 &加密方式

密码变换

**HEX(MD5("0Ui&1m49p9iA5mB2+实际输入的密码+Ix09tY!90ec9U60o"))**

加密方式

**"AES/CFB/NoPadding"，iv="O09&YA)We3qMI5EX "**

## 接口约定

所有取值为"是/否"的参数, 0表示否1表示是;

时间值一律使用64位UTC时间戳表示, 单位为毫秒;

ret\_code 1 表示成功 2表示失败, 如果接口需要具体的错误码在接口定义中声明;

认证参数user\_id & access\_token, 均通过HTTP头传递

# 接口设计

## 认证接口(AUTH)

### 注册验证码&加密key(游客)

|  |  |  |
| --- | --- | --- |
| URL | /flow/auth/signup/keycode | GET |
| 请求 |  | Pts:  1. 注册流程  \* 获取注册SID/验证码/加密key, 注册时回传注册SID/验证码,使用加密key对密码进行加密  2. 注册会话过期时间为1小时  3. |
| 响应 | {  "ret\_code":0 //返回码  "ret\_msg":"xx" //可选，返回错误描述  "ret\_data": {  "signup\_sid":"xxx" // 注册SID(uuid)  "signup\_passwd\_key": "xx" // 密码加密key(长度32)  "signup\_code\_img":"xxx" //注册验证码图片base64  }  } | History:  1. 10-27新增  2. |

### 账号注册(游客)

|  |  |  |
| --- | --- | --- |
| URL | /flow/auth/signup | POST |
| 请求 | {  "signup\_sid ":"xx" // 注册SID  "user\_login ": "xx" // 登录名, 最长30  "passwd\_str": "xx" // 用户密码  "signup\_code":"xx" //注册验证码  } | Pts:  1. 登录名须唯一, 不区分大小写  2. passwd\_str需要变换之后再加密  3. 使用接口1.1返回的加密key对密码进行加密  4. 服务端在解密之后, 将变换后的密码存入数据库  5. |
| 响应 | {  "ret\_code":0 //返回码  //可选，返回错误描述  "ret\_msg":"xx"  } | History:  1. |

### 认证登录(游客)

|  |  |  |
| --- | --- | --- |
| URL | /flow/auth/login | POST |
| 请求 | {  "user\_login": "xx" // 用户名  "login\_auth\_str": "xx" // 认证登录密文  } | Pts:  1. 认证登录密文  **Base64(AES("****32Byte随机密钥+登录名", KEY("变换后的密码")))**  2. 认证结果密文  **Base64(AES("32Byte传输密钥+登录名+Token", KEY("32Byte随机密钥")))**  传输密钥用作后续的接口调用中加密私密信息, Token用作接口用户认证  3. |
| 响应 | {  "ret\_code":0 //返回码  //可选，返回错误描述  "ret\_msg":"xx"  "ret\_data": {  "user\_id": 0L // 用户ID  "auth\_ret\_str":"xx" //认证结果密文串  }  } | History:  1. |

### 重置密码

|  |  |  |
| --- | --- | --- |
| URL | /flow/auth/resetpwd | POST |
| 请求 | {  "user\_id": 0L // 用户ID "access\_token": "xx" // token  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  // 用户新密码  "passwd\_str": "xx"  } | Pts:  1. passwd\_str需要变换之后再加密  2. 使用传输密钥进行加密  3. 服务端在解密之后, 将变换后的密码存入数据库 |
| 响应 | {  "ret\_code":0 //返回码  "ret\_msg":"xx" //可选，返回错误描述  } | History:  1. |

## 基础模块(FND)

### 标签分组(游客)

|  |  |  |
| --- | --- | --- |
| URL | /flow/fnd/tags | POST |
| 请求 | 参数列表：  locale="xx" // 如: zh\_CN, en\_US等 | Pts:  1. 分组标签在系统中用于判断用户感兴趣的主题; 用户可以使用#创建自定义标签  2. 需支持国际化根据不同的locale返回不同语言的数据  3. 标签最大长度为50个字符  4. |
| 响应 | {  "ret\_code":0 // 返回码  "ret\_msg":"xx" // 可选，返回错误描述  "ret\_data":[{  "tag\_group": "xx"  "tag\_list": [{  "tag\_id": 0L  "tag\_text": "xx"  "tag\_hits": 0L  }...]  }...]  } | History:  1. |

### 设置用户标签

|  |  |  |
| --- | --- | --- |
| URL | /flow/fnd/user/tag/update | POST |
| 请求 | {  "user\_id": 0L // 用户ID "access\_token": "xx" // token  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  "tag\_list":[0L, 0L...] // 标签ID列表  } | Pts:  1. |
| 响应 | {  "ret\_code":0 //返回码  "ret\_msg":"xx" //可选，返回错误描述  } | History:  1. |

### 获取用户标签

|  |  |  |
| --- | --- | --- |
| URL | /flow/fnd/user/tag/list | GET |
| 请求 | 参数:  "user\_id" = 0L // 用户ID "access\_token" = "xx" // token | Pts:  1. |
| 响应 | {  "ret\_code":0 //返回码  "ret\_msg":"xx" //可选，返回错误描述  "ret\_data": [{  "tag\_id": 0L  "tag\_text": "xx"  "tag\_hits": 0L  }...]  } | History:  1. |

### 关注/取消

|  |  |  |
| --- | --- | --- |
| URL | /flow/user/follow | POST |
| 请求 | {  "user\_id": 0L // 用户ID "access\_token": "xx" // token  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  "user\_login\_follow": "xx" // 被关注用户  "action\_type":0 // 1 关注 2 取消关注  } | Pts:  1. 做物理删除, 记录日志信息  2. |
| 响应 | {  "ret\_code":0 //返回码  "ret\_msg":"xx" //可选，返回错误描述  } | History:  1. |

### 文件上传

|  |  |  |
| --- | --- | --- |
| URL | /flow/fnd/file/upload | POST |
| 请求 | 文件内容  "user\_id" = 0L // 用户ID "access\_token" = "xx" // token | Pts:  1. 返回的文件url为全路径  2. |
| 响应 | {  "ret\_code":0 //返回码  "ret\_msg":"xx" //可选，返回错误描述  "ret\_data":{  "file\_url":"xx" //文件URL  }  } | History:  1. |

### 未读通知数量

|  |  |  |
| --- | --- | --- |
| URL | /flow/fnd/notice/newcount | GET |
| 请求 | 参数列表：  "user\_id" = 0L // 用户ID "access\_token" = "xx" // token | Pts:  1. |
| 响应 | {  "ret\_code":0 //返回码  "ret\_msg":"xx" //可选，返回错误描述  "ret\_data": {  "new\_count": 0 // 未读消息条数  }  } | History:  1. 新增 - 1023  2. |

### 获取通知列表

|  |  |  |
| --- | --- | --- |
| URL | /flow/fnd/notice/list | GET |
| 请求 | 参数列表：  "user\_id" = 0L // 用户ID "access\_token" = "xx" // token  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  begin=0 // 起止  count=20 // 条数 | Pts:  1. 通知类型  101 ===> 问题@  102 ===> 问题答复  104 ===> 问题评论  105 ===> 答复评论  106 ===> 答复voteup  107 ===> 答复votedown  201 ===> 帖子@  202 ===> 帖子评论  203 ===> 帖子评论@  204 ===> 帖子评论回复  205 ===> 帖子评论回复@  206 ===> 帖子like  207 ===> follow  2. |
| 响应 | {  "ret\_code":0 //返回码  "ret\_msg":"xx" //可选，返回错误描述  "ret\_data": [{  "notice\_id": 0L // id  "notice\_type": 0 // 类型  "notice\_subtype": 0 // 子类型  "notice\_content": "xx" // 内容  "notice\_dtlcontent": "xx" // 详细内容  "attr1": "xx" // attr1  "attr2": "xx" // attr2  "attr3": "xx" // attr3  "attr4": "xx" // attr4  "attr5": "xx" // attr5  "create\_date": 0L // 通知时间  }...]  } | History:  1. 新增 - 1023  2. |

### 更新通知已读

|  |  |  |
| --- | --- | --- |
| URL | /flow/fnd/notice/readall | POST |
| 请求 | {  "user\_id": 0L // 用户ID "access\_token": "xx" // token  } | Pts:  1. |
| 响应 | {  "ret\_code":0 //返回码  "ret\_msg":"xx" //可选，返回错误描述  } | History:  1. 新增 - 1023  2. |

## Discuz模块(DSZ)

### 帖子发布

|  |  |  |
| --- | --- | --- |
| URL | /flow/discuz/post/create | POST |
| 请求 | {  "user\_id": 0L // 用户ID "access\_token": "xx" // token  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  "post\_type": 0 // 帖子类别1基本2 投票  "post\_content": "xx" // 帖子内容; 最长300  "post\_imgs": ["xx1", "xx2"...] // 帖子图片最多三张, 可选  "poll\_setup": { // 投票设置, 可选  "poll\_options": ["xx1", "xx2"...] // 投票选项; 最长50  "poll\_duration": 0L // 投票时间长度  }  } | Pts:  1. 不允许@  2. 异步创建hashtag, 和增加hit  3. hashtag(#)的渲染由客户端进行  4. 投票  \* 进行中  发起人: 可投票/显示投票状况/自己的投票  未投票: 可投票, 不显示投票状况  已投票: 显示投票状况/自己的投票  \* 已结束  显示投票结果/自己的投票 |
| 响应 | {  "ret\_code":0 //返回码  "ret\_msg":"xx" //可选，返回错误描述  "ret\_data": {  "post\_id": 0 // 返回帖子id  "poll\_data": { // 投票数据, 可选  "poll\_id": 0L // 投票ID  "poll\_start ": 0L // 投票开始时间  "poll\_end": 0L // 投票结束时间  "poll\_options": [{  "option\_id": 0L // id  "option\_text": "xx" // 文本  "option\_votes": 0L // 投票数  ]}  }  }  } | History:  1. |

### 帖子删除

|  |  |  |
| --- | --- | --- |
| URL | /flow/discuz/post/delete | POST |
| 请求 | {  "user\_id": 0L // 用户ID "access\_token": "xx" // token  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  "post\_id": 0L // 帖子ID  } | Pts:  1. |
| 响应 | {  "ret\_code":0 //返回码  "ret\_msg":"xx" //可选，返回错误描述  } | History:  1. |

### 点赞/取消

|  |  |  |
| --- | --- | --- |
| URL | /flow/discuz/like | POST |
| 请求 | {  "user\_id": 0L // 用户ID "access\_token": "xx" // token  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  "post\_id": 0L // 帖子id, 可选  "comment\_id": 0L // 评论id, 可选  "action\_type":0 // 1 点赞 2 取消  } | Pts:  1. 做物理删除, 记录日志信息  2. |
| 响应 | {  "ret\_code":0 //返回码  "ret\_msg":"xx" //可选，返回错误描述  } | History:  1. |

### 评论

|  |  |  |
| --- | --- | --- |
| URL | /flow/discuz/comment | POST |
| 请求 | {  "user\_id": 0L // 用户ID "access\_token": "xx" // token  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  "post\_id": 0L // 帖子id, 可选  "comment\_id":0L // 评论id, 可选  "comment\_content": "xx" // 评论内容; 最长300  "comment\_imgs": ["xx1"...] // 评论图片, 最多一张  ~~"mention\_list": ["xx1"...] // @列表; 服务端处理~~  } | Pts:  1. 此接口为回复帖子和回复评论共同使用  2. 异步发送消息提醒: 帖子回复 & mention, 针对目标用户只发送其中一条  3.帖子和评论均是节点(Node), 节点保存路径上所有参与的用户作为metion list, 回复节点时@该节点的mention list  4. 评论thread由各个层级由上往下最早的回复组成; 发表评论后, 首先判断父节点是否存在thread id(tid), 如果有则判断当前节点是否为最先的回复, 如果是则设置为父节点tid, 否则生成新的tid(异步)  5. 评论需保存冗余信息: 帖子id/父评论ID列表/节点level  6. mention(@)由客户端渲染 |
| 响应 | {  "ret\_code":0 //返回码  "ret\_msg":"xx" //可选，返回错误描述  "ret\_data": {  "comment\_id": 0L // 评论id  }  } | History:  1. |

### 帖子详情

|  |  |  |
| --- | --- | --- |
| URL | /flow/discuz/post | POST |
| 请求 | 参数:  "user\_id" = 0L // 用户ID "access\_token" = "xx" // token  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  post\_id = 0L // 帖子id | Pts:  1. |
| 响应 | {  "ret\_code":0 //返回码  "ret\_msg":"xx" //可选，返回错误描述  "ret\_data": {  "post\_type": 0 // 帖子类别1基本2问题3 投票  "post\_content": "xx" // 帖子内容，可选  "post\_imgs": "['xx1'...]" // 帖子图片最多三张, 可选  "poll\_data": { // 投票数据, 可选  "poll\_id": 0L // 投票ID  "poll\_start ": 0L // 投票开始时间  "poll\_end": 0L // 投票结束时间  "poll\_voted": 0 // 用户的投票(option\_id), 可选  "poll\_options": [{  "option\_id": 0L // id  "option\_text": "xx" // 文本  "option\_votes": 0L // 投票数  ]}  }  "comment\_count": 0 // 评论数量  "like\_count": 0 // 点赞数量  "mention\_list": "['xx1'...]" // @列表  "comment\_thread\_list": [ [ {  "comment\_thread": "xx" // 评论thread  "comment\_content": "xx" // 评论内容  "comment\_pics": "['xx1'...]" // 评论图片, 可选  "comment\_count": 0 // 评论数量  "like\_count": 0 // 点赞数量  "metion\_list": "['xx1'...]" // @列表  }... ] ... ]  }  } | History:  1. |

### 评论详情

|  |  |  |
| --- | --- | --- |
| URL | /flow/discuz/comment | POST |
| 请求 | 参数:  "user\_id" = 0L // 用户ID "access\_token" = "xx" // token  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  comment\_id = 0L // 帖子id | Pts:  1. |
| 响应 | {  "ret\_code":0 //返回码  "ret\_msg":"xx" //可选，返回错误描述  "ret\_data": {  // 当前节点thread, 当前节点及以下层级的所有评论  "comment\_thread": [{  "comment\_thread": "xx" // 评论thread  "comment\_content": "xx" // 评论内容  "comment\_pics": "['xx1' ..]" // 评论图片, 可选  "comment\_count": 0 // 评论数量  "like\_count": 0 // 点赞数量  "metion\_list": "['xx1'...]" // @列表  }... ]  // 父评论节点  "ancestor\_comments": [{  "comment\_thread": "xx" // 评论thread  "comment\_content": "xx" // 评论内容  "comment\_pics": "['xx1' ..]" // 评论图片, 可选  "comment\_count": 0 // 评论数量  "like\_count": 0 // 点赞数量  "metion\_list": "['xx1'...]" // @列表  }... ]  // 帖子数据  "post\_data": {  "post\_type": 0 // 帖子类别1基本2问题3 投票  "post\_content": "xx" // 帖子内容，可选  //帖子图片最多三张, 可选  "post\_imgs": "['xx1'...]"  // 投票选项, 可选  "poll\_data": { // 投票数据, 可选  "poll\_id": 0L // 投票ID  "poll\_start ": 0L // 投票开始时间  "poll\_end": 0L // 投票结束时间  "poll\_options": [{  "option\_id": 0L // id  "option\_text": "xx" // 文本  "option\_votes": 0L // 投票数  ]}  }  "comment\_count": 0 // 评论数量  "like\_count": 0 // 点赞数量  "mention\_list": "['xx1'...]" // @列表  }  }  } | History:  1. |

### 帖子列表

|  |  |  |
| --- | --- | --- |
| URL | /flow/discuz/post/list | POST |
| 请求 | 参数:  "user\_id" = 0L // 用户ID "access\_token" = "xx" // token  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  begin: 0 // 起始  count: 0 // 条数 | Pts:  1. 需返回关注人 和 关注tag的帖子  2. |
| 响应 | {  "ret\_code":0 //返回码  "ret\_msg":"xx" //可选，返回错误描述  "ret\_data": [{  "post\_type": 0 // 帖子类别1基本2问题3 投票  "post\_content": "xx" // 帖子内容，可选  "post\_imgs": "['xx1'...]" // 帖子图片最多三张, 可选  "poll\_data": { // 投票数据, 可选  "poll\_id": 0L // 投票ID  "poll\_start ": 0L // 投票开始时间  "poll\_end": 0L // 投票结束时间  "poll\_voted": 0 // 用户的投票(option\_id), 可选  "poll\_options": [{  "option\_id": 0L // id  "option\_text": "xx" // 文本  "option\_votes": 0L // 投票数  ]}  }  "comment\_count": 0 // 评论数量  "like\_count": 0 // 点赞数量  "mention\_list": "['xx1'...]" // @列表  ]...}  } | History:  1. |

### 投票

|  |  |  |
| --- | --- | --- |
| URL | /flow/discuz/poll/vote | POST |
| 请求 | {  "user\_id": 0L // 用户ID "access\_token": "xx" // token  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  "post\_id": 0L // 帖子ID  "option\_id": 0L // 选项ID  } | Pts:  1. 不允许重复投票  2. |
| 响应 | {  "ret\_code":0 //返回码  "ret\_msg":"xx" //可选，返回错误描述  "ret\_data": [{ // 投票数据  "option\_id": 0L // id  "option\_votes": 0L // 投票数  }...]  } | History:  1. |

### 搜索

|  |  |  |
| --- | --- | --- |
| URL | /flow/discuz/post/search | GET |
| 请求 | "user\_id" = 0L // 用户ID "access\_token" = "xx" // token  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  q = "xx" // 搜索关键字 | Pts:  1. 返回结果与2.3.7相同  2. |
| 响应 | 返回帖子列表 | History:  1. |

### 趋势(游客)

|  |  |  |
| --- | --- | --- |
| URL | /flow/discuz/trend | GET |
| 请求 |  | Pts:  1. 根据hashtag的引用统计, 返回一段时间内的趋势  2. |
| 响应 | {  "ret\_code":0 //返回码  "ret\_msg":"xx" //可选，返回错误描述  "ret\_data": [{ // 趋势列表  "hash\_tag": "xx" // hash tag  "tag\_hits": 0L // 次数  }...]  } | History:  1. |

## Tutor模块(TUT)

### 提交教程

|  |  |  |
| --- | --- | --- |
| URL | /flow/tut/submit | POST |
| 请求 | {  "user\_id": 0L // 用户ID "access\_token": "xx" // token  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  "tut\_title": "xx" // 标题; 最长100  "tut\_intro": "xx" // 介绍; 最长300  "tut\_short\_list": "xx" // 简短内容列表; 最长600  "tut\_content\_url": "xx" // 教程内容url, 可以是pdf或者视频  "tut\_attachments": [ { // 教程附件(可选), 允许最多三个  "attach\_title": "xx1", // 附件名称; 最长50  " attach\_url": "xx1" // 附件url  }...]  "tut\_code\_url": "xx1" // 代码地址(可选), 外部地址  "tut\_price": 0.00 // 教程价格(可选), 不传表示免费  } | Pts:  1. 教程内容可以是文件(pdf)或者视频, 支持最多三个附件, 支持外部的代码地址(如: github)  2. |
| 响应 | {  "ret\_code":0 //返回码  "ret\_msg":"xx" //可选，返回错误描述  "ret\_data": {  "tut\_id": 0L // 教程ID  }  } | History:  1. |

### 修改教程

|  |  |  |
| --- | --- | --- |
| URL | /flow/tut/submit | POST |
| 请求 | {  "user\_id": 0L // 用户ID "access\_token": "xx" // token  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  "tut\_id": 0L // 教程id  "tut\_title": "xx" // 标题  "tut\_intro": "xx" // 介绍  "tut\_brief\_list": "xx" // 简单内容列表  "tut\_content\_url": "xx" // 教程内容url, 可以是pdf或者视频  "tut\_attachments": [ { // 教程附件(可选), 允许最多三个  "attach\_id": 0L // 附件ID(可选)  "attach\_title": "xx1" // 附件名称  "attach\_url": "xx1" // 附件url  }...]  "tut\_code\_url": "xx1" // 代码地址(可选), 外部地址  "tut\_price": 0.00 // 教程价格(可选), 不传表示免费  } | Pts:  1. 只有教程发布者能修改自己的教程  2. 所有字段均为必填, 即使没有作修改  3. 如果content url发生了变化, 之前的url进入文件删除列表  4. 附件  \* 当前的附件但不在提交的列表中, 删除该附件并进入删除列表  \* 更新已有的附件时, 如果url发生了变化, 之前的附件进入删除列表  \* 如果attach\_id为空, 表示新建附件  5. |
| 响应 | {  "ret\_code":0 //返回码  "ret\_msg":"xx" //可选，返回错误描述  } | History:  1. |

### 教程点赞/取消

|  |  |  |
| --- | --- | --- |
| URL | /flow/tut/submit | POST |
| 请求 | {  "user\_id": 0L // 用户ID "access\_token": "xx" // token  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  "tut\_id": 0L // 教程id  "action\_type":0 // 1 点赞 2 取消  } | Pts:  1. 做物理删除, 记录日志信息  2. |
| 响应 | {  "ret\_code":0 //返回码  "ret\_msg":"xx" //可选，返回错误描述  } | History:  1. |

### 教程评论

|  |  |  |
| --- | --- | --- |
| URL | /flow/tut/comment | POST |
| 请求 | {  "user\_id": 0L // 用户ID "access\_token": "xx" // token  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  "tut\_id": 0L // 教程ID, 可选  "comment\_id": 0L // 评论ID, 可选  "comment\_content": "xx" // 评论内容, 不超过200字  } | Pts:  1. 评论规则  \* 任何人都可以评论教程, 包括教程发布者  \* thread发起者可以回复该thread  \* 教程发布者可以回复教程的任何评论  2. |
| 响应 | {  "ret\_code":0 //返回码  "ret\_msg":"xx" //可选，返回错误描述  "ret\_data": {  "comment\_id": 0L // 评论ID  }  } | History:  1. |

### 教程详情

|  |  |  |
| --- | --- | --- |
| URL | /flow/tut | GET |
| 请求 | {  "user\_id": 0L // 用户ID "access\_token": "xx" // token  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  "tut\_id": 0L // 教程ID  } | Pts:  1. |
| 响应 | {  "ret\_code":0 //返回码  "ret\_msg":"xx" //可选，返回错误描述  "ret\_data": {  "tut\_title": "xx" // 标题  "tut\_intro": "xx" // 介绍  "tut\_brief\_list": "xx" // 简单内容列表  // 教程内容url, 可以是pdf或者视频  "tut\_content\_url": "xx"  "tut\_attachments": [ { // 教程附件(可选), 允许最多三个  "attach\_id": 0L // 附件ID  "attach\_title": "xx1", // 附件  "attach\_url": "xx1"  }...]  "tut\_code\_url": "xx1" // 代码地址(可选), 外部地址  "tut\_price": 0.00 // 教程价格(可选), 不传表示免费  // thread按照更新时间倒序, 评论按照时间升序  "tut\_comments": [[{  "comment\_thread": "xx" // 评论thread  "comment\_id": 0L // 评论ID  "comment\_user": 0L // 评论者ID  "comment\_content": "xx" // 评论内容  }...]...]  }  } | History:  1. |

## 私信(PRVM)

### 发私信

|  |  |  |
| --- | --- | --- |
| URL | /flow/prvm/mesg/create | POST |
| 请求 | {  "user\_id": 0L // 用户ID "access\_token": "xx" // token  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  "to\_user\_id": 0L // 接收者  "mesg\_type": 0 // 1文字2图片  "mesg\_content": "xx" // 消息内容  } | Pts:  1. |
| 响应 | {  "ret\_code":0 //返回码  "ret\_msg":"xx" //可选, 返回错误描述  "ret\_data": {  "mesg\_id": 0 // 返回消息id  }  } | History:  1. |

### 私信数量

|  |  |  |
| --- | --- | --- |
| URL | /flow/prvm/mesg/count | GET |
| 请求 | 参数：  "user\_id" = 0L // 用户ID "access\_token" = "xx" // token | Pts:  1. 按照thread最后一条消息发送时间倒叙  2. |
| 响应 | {  "ret\_code":0 //返回码  "ret\_msg":"xx" //可选，返回错误描述  "ret\_data": [{  "to\_user\_id":"xx" // 消息接收、发送的对方  "new\_count": 0 // 下行新消息数量  "last\_mesg": "xx" // thread中最后一条消息  "last\_mesg\_type":0 // 最后一条消息类型  "last\_mesg\_date":0L // 最后一条消息发送时间  }... ]  } | History:  1. |

### 私信列表

|  |  |  |
| --- | --- | --- |
| URL | /flow/prvm/mesg/list | GET |
| 请求 | 参数：  "user\_id" = 0L // 用户ID "access\_token" = "xx" // token  "to\_user\_id"=0L // 消息接收、发送的对方  "begin"=0 // 起止  "count"=20 // 条数 | Pts:  1. 分页取最近发送和接收的消息  2. 需标志下行消息为已读  3. |
| 响应 | {  "ret\_code":0 //返回码  "ret\_msg":"xx" //可选，返回错误描述  "ret\_data": [{  "mesg\_type": 0 // 1文字2图片  "mesg\_content":"xx" // 消息内容  "is\_up": 0 // 0 消息下行 1消息上行  "mesg\_status": 0 // 1未读 2已读 对于上行消息  "send\_date":0L // 消息发送时间，以服务端时间为准  } ... ]  } | History:  1. |