后台API手册

**一、特别说明**

1、 这个文档主要是根据《项目后台接口和规范》那个文档中，所定义下来的所有接口而编

写的一个交互样例，主要是为了针对那些接口表达清楚交互的数据详细内容：前端发送哪些具体的数据，后台返回什么样的具体数据。可结合那份文档得到更详细的内容，这份文档可能有遗漏的地方。

2、 返回数据的JSON包含的3个字段的意思：

①code：返回码。0为成功，大于0为失败。如未登录、权限不够等

②msg：错误信息。展示给用户的失败信息。

③detail：请求返回的详细内容。

3、 现在对失败返回码还没有一个详细统一的定义，反正大于0的就是失败，具体错误信息在msg内，不过还不能用于展示给用户，只能做调试用，让前端清楚是什么错误，具体后面再完善错误信息。

4、 ~~另外需要注意的是：现在后台在保持登录这一块，用的是session，前端可以简单的认为是用cookie实现保持登录的~~

~~一个用户，成功登录后，后台会给这个用户分配一个sessionid。后台给前端返回登录成功这个http报文的时候，http头部的cookie域里面是带有sessionid这一项的。前端应该记下这个sessionid，在后续的http请求报文的cookie中，带上这个sessionid项，后台就可以根据这个sessionid判断发出这个请求的用户是否成功登录过了，若未登录会返回错误信息，然后前端就可以跳转到登录界面。~~

~~暂时只有createStory和createChapter这两个接口需要带上sessionid的，其他都是不登录就可以获取的内容。~~

**二、登录接口**

**注意**：在下述样例中qqlogin的access\_token已经失效了，所以不能正常登录，要新的有效的access\_token才行，不过功能肯定是正确的，而sinalogin中的access\_token还是有效的，可以正常使用

**qq登录**

|  |
| --- |
| POST http://127.0.0.1:8000/qqlogin/ |

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 意义 | 备注 | 样例值 |
| access\_token | 前端从qq服务器获得的令牌 | 必传 | CA1CCAE89E13EC710DA88378A40BE5CF |

登录成功的返回结果：

|  |
| --- |
| {  "msg": "",  "code": "0",  "detail": {}  } |

登录失败的返回结果：

|  |
| --- |
| {  "msg": "fail to get openid（out of date access token）",  "code": "1",  "detail": {}  } |

**新浪登录**

|  |
| --- |
| POST http://127.0.0.1:8000/sinalogin/ |

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 意义 | 备注 | 样例值 |
| access\_token | 前端从新浪服务器获得的令牌 | 必传 | 2.008V8iPCeGZxpCb6cb2cbd90rNR1HD |

登录成功的返回结果：

|  |
| --- |
| {  "msg": "",  "code": "0",  "detail": {}  } |

**三、故事/章节（场景）接口**

**创建故事**

|  |
| --- |
| POST http://127.0.0.1:8000/story/create |

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 意义 | 备注 | 样例值 |
| title | 故事标题 | 必传 | abc |
| summary | 故事概要 | 必传 | only a test |
| author | 故事作者 | 必传 | Mingg |
| keysMask | 故事类型的|值（未用） | 必传 | 7 |
| modeMask | 权限类型的|值（未用） | 必传 | 7 |
| startChapDesc | 开始章节的章节概要 | 必传 | start chap’s desc |
| startChapModeMask | 开始章节类型的|值 | 必传 | 7 |

~~http请求头部cookies：~~

|  |  |  |  |
| --- | --- | --- | --- |
| ~~参数~~ | ~~意义~~ | ~~备注~~ | ~~样例值~~ |
| ~~sessionid~~ | ~~登录成功获得的会话id~~ | ~~必传~~ | ~~0td4d5s619jkmrb7s7lpnhzin7rlkrfu~~ |
| ~~Csrf~~ | ~~故事概要~~ | ~~必传~~ | ~~only a test~~ |

创建故事成功的返回结果：

|  |
| --- |
| {  "msg": "",  "code": "0",  "detail": {"stid":"6"}  } |

未登录导致的创建故事失败的返回结果：

|  |
| --- |
| {  "msg": "redirect to login page!",  "code": "1",  "detail": {}  } |

**创建章节**

|  |
| --- |
| POST http://127.0.0.1:8000/chapter/create |

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 意义 | 备注 | 样例值 |
| parentId | 新章节的父亲章节id | 必传 | 19 |
| modeMask | 章节类型的|值（未用） | 必传 | 7 |
| desc | 章节概要 | 必传 | test create new chapter |
| storyId | 章节所属故事的故事id | 必传 | 3 |
| coauthor | 章节作者的id | 必传 | Mingg |

~~http请求头部cookies：~~

|  |  |  |  |
| --- | --- | --- | --- |
| ~~参数~~ | ~~意义~~ | ~~备注~~ | ~~样例值~~ |
| ~~sessionid~~ | ~~登录成功获得的会话id~~ | ~~必传~~ | ~~0td4d5s619jkmrb7s7lpnhzin7rlkrfu~~ |
| ~~Csrf~~ | ~~故事概要~~ | ~~必传~~ | ~~only a test~~ |

创建章节成功的返回结果：

|  |
| --- |
| {  "msg": "",  "code": "0",  "detail": {  "brothers": "21,22,23,24,26,27",  "newcpid": "30"  }  } |

未登录导致的创建章节失败的返回结果：

|  |
| --- |
| {  "msg": "redirect to login page!",  "code": "1",  "detail": {}  } |

**查看故事**

|  |
| --- |
| GET http://127.0.0.1:8000/story/3 |

无附带参数，不需要cookie信息

获取故事成功的返回结果：

|  |
| --- |
| {  "msg": "",  "code": "0",  "detail": {  "startChap": "19",  "scanNum": "0",  "keysMask": "100",  "createTime": "1382083459",  "modeMask": "1",  "title": "abc",  "timeStamp": "1384346257",  "support": "0",  "author": "830B8E22F336682E0989A3F02F0BD8C1",  "summary": "only a test",  "hot": "0", "unsupport": "0",  "collectNum": "0",  "shareNum": "0"  }  } |

获取故事失败的返回结果：

|  |
| --- |
| {  "msg": "story: #3 does not exist",  "code": "1",  "detail": {}  } |

**查看章节**

|  |
| --- |
| GET http://127.0.0.1:8000/chapter/25 |

无附带参数，不需要cookie信息

获取章节成功的返回结果：

|  |
| --- |
| {  "msg": "",  "code": "0",  "detail": {  "storyId": "4",  "unsupport": "0",  "modeMask": "1",  "parentId": "0",  "scanNum": "0",  "support": "0",  "coauthor": "830B8E22F336682E0989A3F02F0BD8C1",  "children": "",  "createTime": "1382321990",  "desc": "start chap"  }  } |

注意：若parentId==0，表示这个章节是一个故事的根章节，没有父亲章节

获取章节失败的返回结果：

|  |
| --- |
| {  "msg": "chapter: #25 does not exist",  "code": "1",  "detail": {}  } |

剩下的故事接口没完成，完成之后再补充对应的接口说明

**四、榜单接口**

注意：其实都是用/storyList/{type}/{start}/{count}/{timeStamp}/{startStoryId}这个接口，不过随着url携带参数的变化，有下面4种请求。

关于start和count参数的说明：

譬如有5个故事，它们按照时间排序（或者根据热度排序，与type参数有关系）的结果是：3,2,4,1,5；那么如果start==3，count==2，那么后台会返回的故事是4和1.

**请求最新故事列表（不使用startStoryId）**

|  |
| --- |
| GET http://127.0.0.1:8000/storyList/1/1/30/1382325817/-1 |

注意：若不使用startStoryId，在url中其对应参数需要为-1

无附带参数，不需要cookie信息

获取最新故事列表成功的返回结果：

|  |
| --- |
| {  "msg": "",  "code": "0",  "detail": {  "stories": [  {  "startChap": "29",  "modeMask": "1",  "title": "abc",  "timeStamp": "1384345924",  "support": "0",  "author": "2064502873",  "modify": "1",  "summary": "only a test",  "hot": "0",  "unsupport": "0",  "keysMask": "100",  "storyid": "6",  "collectNum": "0",  "shareNum": "0",  "createTime": "1384345924",  "scanNum": "0"  },  {  "modify": "0",  "storyid": "4"  }  ]  }  } |

特别注意

1. 特别注意每个故事中的modify字段 ，若modify字段值为1，那么就会有这个故事的详细信息（因为通过时间戳判断，这个故事没有被更新过）；若modify字段为0，那么不会有这个故事的详细信息。
2. stories里面存有的故事个数不一定等于url请求中的count，因为可能不够count个故事了，譬如这个例子，请求30个故事，可是只返回两个，因为只有两个故事。

**请求最新故事列表（使用startStoryId）**

|  |
| --- |
| GET http://127.0.0.1:8000/storyList/1/1/30/1382325817/3 |

注意：startStoryId不为-1，表示指定后台返回从故事编号为3的故事后面的最新count个故事，包括故事id为3的这个故事

无附带参数，不需要cookie信息

获取最新故事列表成功的返回结果：

|  |
| --- |
| {  "msg": "",  "code": "0",  "detail": {  "stories": [  {  "startChap": "29",  "modeMask": "1",  "title": "abc",  "timeStamp": "1384345924",  "support": "0",  "author": "2064502873",  "modify": "1",  "summary": "only a test",  "hot": "0",  "unsupport": "0",  "keysMask": "100",  "storyid": "3",  "collectNum": "0",  "shareNum": "0",  "createTime": "1384345924",  "scanNum": "0"  },  {  "modify": "0",  "storyid": "4"  }  ]  }  } |

特别注意

1. 特别注意每个故事中的modify字段 ，若modify字段值为1，那么就会有这个故事的详细信息（因为通过时间戳判断，这个故事没有被更新过）；若modify字段为0，那么不会有这个故事的详细信息。
2. stories里面存有的故事个数不一定等于url请求中的count，因为可能不够count个故事了！

**请求最热故事列表**

|  |
| --- |
| GET http://127.0.0.1:8000/storyList/2/1/30/1382325817/-1 |

无附带参数，不需要cookie信息

获取最新故事列表成功的返回结果：

|  |
| --- |
| {  "msg": "",  "code": "0",  "detail": {  "stories": [  {  "startChap": "29",  "modeMask": "1",  "title": "abc",  "timeStamp": "1384345924",  "support": "0",  "author": "2064502873",  "modify": "1",  "summary": "only a test",  "hot": "0",  "unsupport": "0",  "keysMask": "100",  "storyid": "3",  "collectNum": "0",  "shareNum": "0",  "createTime": "1384345924",  "scanNum":" 0"  },  {  "modify": "0",  "storyid": "4"  }  ]  }  } |

特别注意

1. 特别注意每个故事中的modify字段 ，若modify字段值为1，那么就会有这个故事的详细信息（因为通过时间戳判断，这个故事没有被更新过）；若modify字段为0，那么不会有这个故事的详细信息。
2. stories里面存有的故事个数不一定等于url请求中的count，因为可能不够count个故事了！

**请求精品故事列表**

|  |
| --- |
| GET http://127.0.0.1:8000/storyList/3/1/30/1382325817/-1 |

无附带参数，不需要cookie信息

获取最新故事列表成功的返回结果：

|  |
| --- |
| {  "msg": "",  "code": "0",  "detail": {  "stories": [  {  "startChap": "29",  "modeMask": "1",  "title": "abc",  "timeStamp": "1384345924",  "support": "0",  "author": "2064502873",  "modify": "1",  "summary": "only a test",  "hot": "0",  "unsupport": "0",  "keysMask": "100",  "storyid": "3",  "collectNum": "0",  "shareNum": "0",  "createTime": "1384345924",  "scanNum": "0"  },  {  "modify": "0",  "storyid": "4"  }  ]  }  } |

特别注意

1. 特别注意每个故事中的modify字段 ，若modify字段值为1，那么就会有这个故事的详细信息（因为通过时间戳判断，这个故事没有被更新过）；若modify字段为0，那么不会有这个故事的详细信息。
2. stories里面存有的故事个数不一定等于url请求中的count，因为可能不够count个故事了！

统计接口和更新接口还没完成，后面完成再补充相关的交互说明。