

“酒井实践在线” API 文档

API 文档是开发者的协作指南，通过清晰定义接口功能、用法，确保团队在开发过程中保持一致。

本文档需要列出软件后端服务提供的所有 API，并按照合理的结构进行组织（可按模块 / 资源类型划分）。

用户信息操作

发送验证邮件

[POST] /api/auth/send-email/

向用户填写的邮箱中发送验证密码

示例

请求

代码块

```
1  {
2      "email": "test@example.com",
3  }
```

响应

代码块

```
1  {
2      "message": "验证码已发送到您的邮箱，请查收。"
3  }
```

请求参数

参数	类型	说明
email	str	接受验证码的邮箱

响应状态

返回提示	说明
message	操作成功
error	存在问题（具体问题由包含的内容给出）

用户注册

[POST] /api/auth/register/

处理用户注册

示例

请求

代码块

```
1  {
2      "username": "user",
3      "email": "test@example.com",
4      "student_id": "123456",
5      "phone_number": "13900000000",
6      "password": "password",
7      "verification_code": "123456"
8  }
```

响应

代码块

```
1  {
2      "message": "注册成功"
3  }
```

请求参数

参数	类型	说明
username	str	用户名
email	str	邮箱
student_id	str	学号
phone_number	str	手机号
password	str	密码
verification_code	str	验证码

响应状态

返回提示	说明
message	操作成功
error	存在问题（具体问题由包含的内容给出）

用户登录

[POST] /api/auth/login/

处理用户登录

示例

请求

代码块

```
1  {
2      "username": "user",
3      "student_id": "123456",
4      "phone_number": "13900000000",
```

```
5     "password": "password",
6 }
```

响应

代码块

```
1  {
2      "message": "登录成功",
3      "user_info": {
4          "id": 2,
5          "username": "wyk_back",
6          "email": "wangyukai@mails.tsinghua.edu.cn",
7          "student_id": null,
8          "phone_number": null,
9          "user_permission": "超级管理员"
10     }
11 }
```

请求数据

参数	类型	说明
username	str	用户名
student_id	str	学号
phone_number	str	手机号
password	str	密码

响应状态

返回提示	说明
message	操作成功
error	存在问题（具体问题由包含的内容给出）

响应数据

参数	类型	说明
id	int	数据库主键
username	str	用户名
email	str	邮箱
student_id	str	学号
phone_number	str	手机号
user_permission	str	用户权限

绑定飞书账号

[GET] api/auth/feishu-bind/

示例

响应：

代码块

```
1  {
2      "auth_url": "....."
3  }
```

响应状态

参数	说明
error	存在问题（具体问题由包含的内容给出）
无	正常返回

响应数据

参数	类型	说明
auth_url	str	飞书绑定的跳转url

获取用户信息

[GET] api/auth/user-info

示例

响应：

代码块

```
1  {
2      "message": "用户已登录",
3      "user-info": {
4          "id": 2,
5          "username": "wyk_back",
6          "email": "wangyukai@mails.tsinghua.edu.cn",
7          "student_id": null,
8          "phone_number": null,
9          "user_permission": "超级管理员"
10     }
11 }
```

响应状态

返回提示	说明
message	操作成功
error	存在问题（具体问题由包含的内容给出）

响应数据

参数	类型	说明
id	int	数据库主键
username	str	用户名
email	str	邮箱
student_id	str	学号
phone_number	str	手机号
user_permission	str	用户权限

根据id获取用户信息

[POST] api/user/get-user-by-id/

示例

请求

代码块

```
1  {
2      "id": 1
3  }
```

响应

代码块

```
1  {
2      "user-info":{
3          "id": 2,
4          "username": "wyk_back",
5          "email": "wangyukai@mails.tsinghua.edu.cn",
6          "student_id": null,
7          "phone_number": null,
8          "user_permission": "超级管理员"
9      }
10 }
```

响应状态

返回提示	说明
message	操作成功
error	存在问题（具体问题由包含的内容给出）

响应数据

参数	类型	说明

id	int	数据库主键
username	str	用户名
email	str	邮箱
student_id	str	学号
phone_number	str	手机号
user_permission	str	用户权限

用户权限管理

查询所有用户

[GET] api/user/get-user/?page=1

示例

响应：

代码块

```
1  {
2      "count": 100,
3      "next": "http://localhost:8000/api/users/?page=2",
4      "previous": null,
5      "results": [
6          {
7              "id": 1,
8              "username": "john_doe",
9              "email": "john@example.com",
10             "student_id": "20230001",
11             "phone_number": "13800138000",
12             "user_permission": "普通用户"
13         },
14         {
15             "id": 2,
16             "username": "admin",
17             "email": "admin@example.com",
18             "student_id": null,
19             "phone_number": null,
20             "user_permission": "超级管理员"
21         }
22     ]
23 }
```

响应状态

参数	说明
error	存在问题（具体问题由包含的内容给出）
无	正常返回

响应数据

--	--	--

参数	类型	说明
count	int	数据数量
next	str	下一页的链接
previous	str	上一页的链接
results	list	结果列表
id	int	数据库主键
username	str	用户名
email	str	邮箱
student_id	str	学号
phone_number	str	手机号
user_permission	str	用户权限

修改权限

[POST] api/user/modify-permission/

示例

请求：

前端发送用户名和将要修改为的权限

代码块

```
1  {
2      "username": "user",
3      "user_permission": "普通管理员",
4  }
```

响应：

代码块

```
1  {
2      "message" : "成功修改权限",
3  }
```

请求参数

参数	类型	说明
username	str	修改的用户名
user_permission	str	要修改成的用户权限

响应状态

返回提示	说明
message	操作成功

error	存在问题（具体问题由包含的内容给出）
-------	--------------------

查询所有普通管理员

[GET] api/user/get-administrator/?page=1

示例

响应：

代码块

```
1  {
2      "count": 100,
3      "next": "http://localhost:8000/api/users/?page=2",
4      "previous": null,
5      "results": [
6          {
7              "id": 1,
8              "username": "john_doe",
9              "email": "john@example.com",
10             "student_id": "20230001",
11             "phone_number": "13800138000",
12             "user_permission": "普通管理员"
13         },
14         {
15             "id": 2,
16             "username": "admin",
17             "email": "admin@example.com",
18             "student_id": null,
19             "phone_number": null,
20             "user_permission": "普通管理员"
21         }
22     ]
23 }
```

响应状态

参数	说明
error	存在问题（具体问题由包含的内容给出）
无	正常返回

响应数据

参数	类型	说明
count	int	数据数量
next	str	下一页的链接
previous	str	上一页的链接
results	list	结果列表
id	int	数据库主键
username	str	用户名

email	str	邮箱
student_id	str	学号
phone_number	str	手机号
user_permission	str	用户权限

查询所有超级管理员

[GET] api/user/get-super/?page=1

示例：

响应

代码块

```
1  {
2      "count": 100,
3      "next": "http://localhost:8000/api/users/?page=2",
4      "previous": null,
5      "results": [
6          {
7              "id": 1,
8              "username": "john_doe",
9              "email": "john@example.com",
10             "student_id": "20230001",
11             "phone_number": "13800138000",
12             "user_permission": "超级管理员"
13         },
14         {
15             "id": 2,
16             "username": "admin",
17             "email": "admin@example.com",
18             "student_id": null,
19             "phone_number": null,
20             "user_permission": "超级管理员"
21         }
22     ]
23 }
```

响应状态

参数	说明
error	存在问题（具体问题由包含的内容给出）
无	正常返回

响应数据

参数	类型	说明
count	int	数据数量
next	str	下一页的链接
previous	str	上一页的链接

results	list	结果列表
id	int	数据库主键
username	str	用户名
email	str	邮箱
student_id	str	学号
phone_number	str	手机号
user_permission	str	用户权限

查询所有管理员（包含普通和超级管理员）

[GET] api/user/get-all-administrator/?page=1

示例

响应：

代码块

```
1  "all-admin": [  
2    {  
3      "id": 1,  
4      "username": "john_doe",  
5      "email": "john@example.com",  
6      "student_id": "20230001",  
7      "phone_number": "13800138000",  
8      "user_permission": "普通管理员"  
9    },  
10   {  
11     "id": 2,  
12     "username": "admin",  
13     "email": "admin@example.com",  
14     "student_id": null,  
15     "phone_number": null,  
16     "user_permission": "超级管理员"  
17   }  
18 ]
```

响应状态

参数	说明
error	存在问题（具体问题由包含的内容给出）
无	正常返回

响应数据

参数	类型	说明
count	int	数据数量
next	str	下一页的链接

previous	str	上一页的链接
results	list	结果列表
id	int	数据库主键
username	str	用户名
email	str	邮箱
student_id	str	学号
phone_number	str	手机号
user_permission	str	用户权限

创建支队

创建支队

[POST] api/detachment/create/

示例

请求：

代码块

```
1  {
2      "name" : "testdetachment",
3      "start_date" : "2025-04-13",
4      "end_date" : "2025-04-14",
5      "detachment_leader" : ["username1", "username2", ...],
6      "detachment_member" : ["username1", "username2", ...],
7  }
```

响应：

代码块

```
1  {
2      "message" : "成功创建支队",
3  }
```

请求参数

参数	类型	说明
name	str	支队名
start_date	str	开始日期
end_date	str	结束日期
detachment_leader	list	支队长名单
detachment_member	list	支队员名单

响应状态

返回提示	说明
message	操作成功
error	存在问题（具体问题由包含的内容给出）

修改支队信息

[POST] api/detachment/modify/

示例

请求：

代码块

```
1  {
2      "id" : 1,
3      "name" : "testdetachment",
4      "start_date" : "2025-04-13",
5      "end_date" : "2025-04-14",
6      "detachment_member" : ["username1", "username2", ...],
7  }
```

响应：

代码块

```
1  {
2      "message" : "成功修改支队信息",
3  }
```

请求参数

参数	类型	说明
id	int	修改的支队主键
name	str	修改后的支队名
start_date	str	修改后的开始日期
end_date	str	修改后的结束日期
detachment_member	list	修改后的队员列表

响应状态

返回提示	说明
message	操作成功
error	存在问题（具体问题由包含的内容给出）

停用支队

[POST] api/detachment/deactivate/

示例

请求：

代码块

```
1  {
2      "id" : 1,
3  }
```

响应：

代码块

```
1  {
2      "message" : "成功停用支队",
3  }
```

请求参数

参数	类型	说明
id	int	停用的支队主键

响应状态

返回提示	说明
message	操作成功
error	存在问题（具体问题由包含的内容给出）

删除支队

[POST] api/detachment/delete/

示例

请求：

代码块

```
1  {
2      "id" : 1,
3  }
```

响应：

代码块

```
1  {
2      "message" : "成功删除支队",
3  }
```

请求参数

参数	类型	说明
id	int	删除的支队主键

响应状态

返回提示	说明
message	操作成功
error	存在问题（具体问题由包含的内容给出）

获取全部支队

[GET] api/detachment/get-all/?page=1

示例

响应：

代码块

```
1  {
2      "count": 25,                // 总通知数量
3      "next": "http://api.example.com/detachments/?page=2", // 下一页链接
4      "previous": null,          // 上一页链接（第一页时为null）
5      "results" : [{
6          "id" : 1,
7          "name" : "testdetachment",
8          "start_date" : "2025-04-13",
9          "end_date" : "2025-04-14",
10         "detachment_leader" : ["username1", "username2", ...],
11         "detachment_member" : ["username1", "username2", ...],
12         "valid" : True / False
13     }, ...], // 所有支队
14 }
```

响应状态

参数	说明
error	存在问题（具体问题由包含的内容给出）
无	正常返回

响应数据

参数	类型	说明
count	int	数据数量
next	str	下一页的链接
previous	str	上一页的链接
results	list	结果列表
id	int	数据库主键

name	str	支队名
start_date	str	开始日期
end_date	str	结束日期
detachment_leader	list	支队长名单
detachment_member	list	支队员名单
valid	bool	是否有效

获取所有有效支队

[GET] api/detachment/get-valid/?page=1

示例

响应：

代码块

```
1  {
2      "count": 25,                // 总通知数量
3      "next": "http://api.example.com/detachments/?page=2",  // 下一页链接
4      "previous": null,          // 上一页链接（第一页时为null）
5      "results" : [{
6          "id" : 1,
7          "name" : "testdetachment",
8          "start_date" : "2025-04-13",
9          "end_date" : "2025-04-14",
10         "detachment_leader" : ["username1", "username2", ...],
11         "detachment_member" : ["username1", "username2", ...],
12         "valid" : True / False
13     }, ...],  // 所有支队
14 }
```

响应状态

参数	说明
error	存在问题（具体问题由包含的内容给出）
无	正常返回

响应数据

参数	类型	说明
count	int	数据数量
next	str	下一页的链接
previous	str	上一页的链接
results	list	结果列表
id	int	数据库主键

name	str	支队名
start_date	str	开始日期
end_date	str	结束日期
detachment_leader	list	支队长名单
detachment_member	list	支队员名单
valid	bool	是否有效

发布通知公告

管理员发布通知界面

[POST] api/notice/send-notice/

示例

请求：

代码块

```
1  {
2      "title" : ".....", // 公告标题
3      "content" : ".....", // 通知内容
4      "sender" : "团委实践组", // 发布者（默认为团委实践组）
5      "detachment" : ["key_1", "key_2", ....] // 发送给哪些支队
6  }
```

响应：

代码块

```
1  {
2      "message" : "发送成功",
3  }
```

请求参数

参数	类型	说明
title	str	通知标题
content	str	通知内容
sender	str	发送者
detachment	list	接受的支队列表

响应状态

返回提示	说明
message	操作成功
error	存在问题（具体问题由包含的内容给出）

支队长查询收到的通知

[GET] api/notice/get-notice/

示例

响应：

代码块

```
1  {
2      "count": 25,                // 总通知数量
3      "next": "http://api.example.com/notices/?page=2", // 下一页链接
4      "previous": null,          // 上一页链接（第一页时为null）
5      "results": [
6          {
7              "id": 1,            // 通知ID（来自关联的Notice模型）
8              "title": "系统维护通知", // 通知标题
9              "content": "将于今晚23:00进行系统升级...", // 通知内容
10             "sender": "管理员", // 发送者名称
11             "confirmed": false // 用户是否已确认（来自UserNotice模型）
12         },
13         {
14             "id": 2,
15             "title": "订单发货提醒",
16             "content": "您的订单已由顺丰快递发出...",
17             "sender": "物流部",
18             "confirmed": true
19         }
20     ]
21 }
22
```

响应状态

参数	说明
error	存在问题（具体问题由包含的内容给出）
无	正常返回

响应数据

参数	类型	说明
count	int	数据数量
next	str	下一页的链接
previous	str	上一页的链接
results	list	结果列表
id	int	通知主键
title	str	通知标题
content	str	通知内容
sender	str	发送者

confirmed	bool	是否确认
-----------	------	------

支队长确认通知

[POST] api/notice/confirm

示例

请求：

```
代码块
1  {
2      "id" : 1, //主键
3  }
```

响应：

```
代码块
1  {
2      "message" : "确认成功",
3  }
```

请求参数

参数	类型	说明
id	int	通知主键

响应状态

返回提示	说明
message	操作成功
error	存在问题（具体问题由包含的内容给出）

管理员查询所有通知

[GET] api/notice/query/?page=1

示例

响应：

```
代码块
1  {
2      "count": 2,           // 总记录数
3      "next": null,        // 下一页链接（此处为 null 表示最后一页）
4      "previous": null,    // 上一页链接
5      "results": [         // 当前页数据（每页 20 条）
6          {
7              "id": 1,
8              "title": "暑期实践通知",
9              "content": "请各支队于 10 月前提交材料...",
10             "sender": "团委实践组",
```

```
11     "detachment": [1, 3],    // 关联支队的 ID 列表
12     "date": "2023-09-20"    // 自动生成的日期
13 },
14 {
15     "id": 2,
16     "title": "安全培训提醒",
17     "content": "本周五下午进行安全培训...",
18     "sender": "团委实践组",
19     "detachment": [2, 5],
20     "date": "2023-09-22"
21 }
22 ]
23 }
```

响应状态

参数	说明
error	存在问题（具体问题由包含的内容给出）
无	正常返回

响应数据

参数	类型	说明
count	int	数据数量
next	str	下一页的链接
previous	str	上一页的链接
results	list	结果列表
id	int	通知主键
title	str	通知标题
content	str	通知内容
sender	str	发送者
detachment	list	接受的支队主键
date	str	发送日期

管理员查询某条通知的确认情况

[POST] api/notice/query-confirm

示例

请求：

代码块

```
1  {
2    "id" : 1, //主键
3  }
```

响应：

代码块

```
1  {
2      "title" : "....", // 查询的通知标题
3      "all_user" : ["name_1", "name_2", ....], // 通知发送给的所有支队名
4      "confirmed_user" : ["name_1", "name_2", ....], // 当前已经确认的支队名
5  }
```

请求参数

参数	类型	说明
id	int	通知主键

响应状态

参数	说明
error	存在问题（具体问题由包含的内容给出）
无	正常返回

响应数据

参数	类型	说明
title	str	通知标题
all_user	list	所有接收支队
confirmed_user	list	已经确认的支队

外联清单管理

上传Excel更新外联清单

[POST] api/connection-list/upload/

示例

请求：

一个Excel文件

响应：

代码块

```
1  {
2      "message" : "上传成功",
3  }
```

响应状态

返回提示	说明
message	操作成功
error	存在问题（具体问题由包含的内容给出）

查询外联清单

[GET] api/connection-list/query/?page=1

示例

响应：

代码块

```
1  {
2    "count": 2,           // 总记录数
3    "next": null,        // 下一页链接（此处为 null 表示最后一页）
4    "previous": null,    // 上一页链接
5    "results": [         // 当前页数据（每页 20 条）
6      {
7        "id": 1,
8        "detachment_name" : "数蜀拓新", // 支队名
9        "leader" : "王誉凯",           // 支队长
10       "theme" : "探究东数西算",      // 实践主题
11       "duration" : "2025年1月1日至1月2日", // 出行时间，格式不固定，字符串直接展示即可
12       "location" : "四川成都",       // 实践地
13       "enterprise" : "企业1、企业2", // 参访企业，格式不固定，字符串直接展示即可
14       "government" : "政府1, 政府2", // 政府机构，同上
15       "venue" : "场馆1, 场馆2",     // 场馆，同上
16     },
17     {
18       "id": 2,
19       "detachment_name" : "数蜀拓新",
20       "leader" : "王誉凯",
21       "theme" : "探究东数西算",
22       "duration" : "2025年1月1日至1月2日",
23       "location" : "四川成都",
24       "enterprise" : "企业1、企业2",
25       "government" : "政府1, 政府2",
26       "venue" : "场馆1, 场馆2",
27     }
28   ]
29 }
```

响应状态

参数	说明
error	存在问题（具体问题由包含的内容给出）
无	正常返回

响应数据

参数	类型	说明
count	int	数据数量

next	str	下一页的链接
previous	str	上一页的链接
results	list	结果列表
id	int	外联条目主键
detachment_name	str	支队名
leader	str	支队长
theme	str	实践主题
duration	str	出行时间
location	str	出行地点
enterprise	str	参访企业
government	str	参访的政府机构
venue	str	参访场馆

清空外联清单

[POST] api/connection-list/clear/

示例

响应：

代码块

```
1  {
2      "message" : "清空成功",
3  }
```

响应状态

返回提示	说明
message	操作成功
error	存在问题（具体问题由包含的内容给出）

下载外联清单

[POST] api/connection-list/download/

示例

响应：

外联清单的Excel文件

响应状态

参数	说明

error	存在问题（具体问题由包含的内容给出）
无	正常返回

响应数据

由外联清单导出的Excel文件

推送审核

支队长送审

[POST] api/approval/send/

示例

请求

代码块

```
1  {
2      "link" : "https://www.xiumi.com/....." // 推送的秀米链接
3      "sender_name" : "王誉凯" // 送审人姓名
4  }
```

响应：

代码块

```
1  {
2      "message" : "操作成功",
3  }
```

请求参数

参数	类型	说明
link	str	推送链接
sender_name	str	送审人姓名

响应状态

返回提示	说明
message	操作成功
error	存在问题（具体问题由包含的内容给出）

管理员查询待审核的推送

[GET] api/approval/query/?page=1

示例

响应：

--

```
1 代码块
2      "count": 1,           // 总记录数
3      "next": null,        // 下一页链接（此处为 null 表示最后一页）
4      "previous": null,    // 上一页链接
5      "results": [         // 当前页数据（每页 20 条）
6          {
7              "id": 1,      // 审核模型的主键
8              "sender_name" : "王誉凯", // 送审人
9              "link" : "https://www.xiumi.com/..." // 推送链接
10         },
11     ]
12 }
```

响应状态

参数	说明
error	存在问题（具体问题由包含的内容给出）
无	正常返回

响应数据

参数	类型	说明
count	int	数据数量
next	str	下一页的链接
previous	str	上一页的链接
results	list	结果列表
id	int	推送主键
link	str	推送链接
sender_name	str	送审人姓名

管理员审核推送通过

[POST] api/approval/pass-down/

示例

请求：

```
代码块
1  {
2      "id" : 1, // 审核模型的主键
3  }
```

响应：

```
代码块
1  {
2      "message" : "操作成功",
```



```
3    }
```

请求参数

参数	类型	说明
id	int	推送主键

响应状态

返回提示	说明
message	操作成功
error	存在问题（具体问题由包含的内容给出）

管理员审核推送不通过

[POST] api/approval/reject/

示例

请求：

代码块

```
1  {
2      "id" : 1, // 审核模型的主键
3      "message" : "排版有问题", // 评语
4  }
```

响应：

代码块

```
1  {
2      "message" : "操作成功",
3  }
```

请求参数

参数	类型	说明
id	int	推送主键
message	str	推送的问题

响应状态

返回提示	说明
message	操作成功
error	存在问题（具体问题由包含的内容给出）

管理员决定推送可以发表

[POST] api/approval/approve/

示例

请求：

代码块

```
1  {
2      "id" : 1 // 允许发表的审核模型主键
3  }
```

响应：

代码块

```
1  {
2      "message" : "操作成功",
3  }
```

请求参数

参数	类型	说明
id	int	推送主键

响应状态

返回提示	说明
message	操作成功
error	存在问题（具体问题由包含的内容给出）

支队长查询审核状态

[GET] api/approval/query-status/?page=1

示例

响应：

代码块

```
1  {
2      "count": 1,           // 总记录数
3      "next": null,        // 下一页链接（此处为 null 表示最后一页）
4      "previous": null,    // 上一页链接
5      "results": [         // 当前页数据（每页 20 条）
6          {
7              "id": 1,      // 审核模型的主键
8              "sender_name" : "王誉凯", // 送审人
9              "link" : "https://www.xiumi.com/..." // 推送链接
10             "status" : "审核中 / 未通过 / 已通过",
11             "message" : "" / "排版有问题" // 评语，可能为空字符串
12             "reviewer" : {
```

```
13         "username" : "administrator", // 审核人用户名
14     }
15 },
16 ]
17 }
```

响应状态

参数	说明
error	存在问题（具体问题由包含的内容给出）
无	正常返回

响应数据

参数	类型	说明
count	int	数据数量
next	str	下一页的链接
previous	str	上一页的链接
results	list	结果列表
id	int	推送主键
sender_name	str	送审人姓名
link	str	推送链接
status	str	审核状态
message	str	审核评语
reviewer.username	str	审核人用户名

支队长修改审核的推送链接

[POST] api/approval/modify/

示例

请求：

代码块

```
1  {
2      "id" : 1, // 审核模型的主键
3      "link" : "https://www.xiumi.com/...." // 推送链接
4  }
```

响应：

代码块

```
1  {
2      "message" : "操作成功",
```

```
3    }
```

请求参数

参数	类型	说明
id	int	推送主键
link	str	修改后的推送链接

响应状态

参数	说明
error	存在问题（具体问题由包含的内容给出）
无	正常返回

超级管理员管理审核流程

[POST] api/approval/manage/

示例

请求：

代码块

```
1  {
2      "user-id" : [1,3,5,7,9,...], // 列表前面的先审核，再给后面的
3      "username" : ["administrator_1", "administrator_2", ...] // id和用户名有一个即可，都有的话优先id
4  }
```

响应：

代码块

```
1  {
2      "message" : "操作成功",
3  }
```

请求参数

参数	类型	说明
user-id	list	审核人的主键列表
username	list	审核人的用户名列表

响应状态

返回提示	说明
message	操作成功

error	存在问题（具体问题由包含的内容给出）
-------	--------------------

展示审核流程

[GET] api/approval/query-reviewer/

示例

响应：

代码块

```
1  {
2      "count": 2,
3      "next": null,
4      "previous": null,
5      "results": [
6          {
7              "order": 2, // 第几个审核人
8              "reviewer": { // 审核人信息
9                  "id": 4,
10                 "username": "zhouzk23",
11                 "email": "zhouzk23@mails.tsinghua.edu.cn",
12                 "student_id": "2023010774",
13                 "phone_number": null,
14                 "user_permission": "超级管理员"
15             }
16         },
17         {
18             "order": 1,
19             "reviewer": {
20                 "id": 2,
21                 "username": "wyk_back",
22                 "email": "wangyukai@mails.tsinghua.edu.cn",
23                 "student_id": null,
24                 "phone_number": null,
25                 "user_permission": "超级管理员"
26             }
27         }
28     ]
29 }
```

响应状态

参数	说明
error	存在问题（具体问题由包含的内容给出）
无	正常返回

响应数据

参数	类型	说明
count	int	数据数量
next	str	下一页的链接
previous	str	上一页的链接

results	list	结果列表
order	int	审核顺序
id	int	用户主键
username	str	用户名
email	str	邮箱
student_id	str	学号
phone_number	str	手机号
user_permission	str	用户权限

支队长手册

获取支队长手册链接

[GET] api/handbook/get-link/

示例

响应：

代码块

```
1  {
2      "handbook": [
3          {
4              "id": 1,
5              "document_id": "NPY3dICP0oHZA7xLV6qcBSaMnmc",
6              "url": "https://xcnck9sczoe4.feishu.cn/docx/NPY3dICP0oHZA7xLV6qcBSaMnmc",
7              "title": "支队长手册",
8              "created_at": "2025-05-09T12:44:28.340494"
9          }
10     ]
11 }
```

响应状态

参数	说明
error	存在问题（具体问题由包含的内容给出）
无	正常返回

响应数据

参数	类型	说明
id	int	手册主键
document_id	str	手册在飞书端的id
url	str	手册链接
title	str	手册标题

created_at	str	创建时间
------------	-----	------

创建支队长手册

[POST] api/handbook/create/

示例

请求：

代码块

```
1  {
2      "title": "....."
3  }
```

响应：

代码块

```
1  {
2      "url": "....."
3  }
```

请求参数

参数	类型	说明
title	str	手册标题

响应状态

参数	说明
error	存在问题（具体问题由包含的内容给出）
无	正常返回

响应数据

参数	类型	说明
url	str	飞书云文档的url

获取编辑权限

[POST] api/handbook/add-coauthor/

示例

请求：

代码块

```
1  {
2      "url": "....."
}
```

```
3    }
```

响应：

```
代码块
1    {
2        "message": "您已成为文档协作者"
3    }
```

请求参数

参数	类型	说明
url	str	飞书云文档的url

响应状态

返回提示	说明
message	操作成功
error	存在问题（具体问题由包含的内容给出）

删除支队长手册

[POST] api/handbook/delete-handbook/

示例

请求：

```
代码块
1    {
2        "url": "....."
3    }
```

响应：

```
代码块
1    {
2        "message" : "操作成功",
3    }
```

请求参数

参数	类型	说明
url	str	飞书云文档的url

响应状态

返回提示	说明
message	操作成功
error	存在问题（具体问题由包含的内容给出）

修改标题

[POST] api/handbook/modify-title

示例

请求：

代码块

```
1  {
2      "url": ".....",
3      "title": "新的标题"
4  }
```

响应：

代码块

```
1  {
2      "message" : "操作成功",
3  }
```

请求参数

参数	类型	说明
url	str	飞书云文档的url
title	str	新的标题

响应状态

返回提示	说明
message	操作成功
error	存在问题（具体问题由包含的内容给出）

公函审批

管理员上传公函模板

[POST] api/letter/upload-template/

示例

请求：

一个word或pdf文件

响应：

代码块

```
1  {
2      "message" : "上传成功",
3  }
```

请求参数

一个word或pdf文件

响应状态

返回提示	说明
message	操作成功
error	存在问题（具体问题由包含的内容给出）

查询所有的公函模板

[GET] api/letter/get-template/

示例

响应：

代码块

```
1  {
2      "count": 1,           // 总记录数
3      "next": null,        // 下一页链接（此处为 null 表示最后一页）
4      "previous": null,    // 上一页链接
5      "results": [         // 当前页数据（每页 20 条）
6          {
7              "id": 1,      // 主键
8              "filename" : "介绍信模板.docx" // 文件名
9          },
10     ]
11 }
```

响应状态

参数	说明
error	存在问题（具体问题由包含的内容给出）
无	正常返回

响应数据

参数	类型	说明
count	int	数据数量
next	str	下一页的链接
previous	str	上一页的链接

results	list	结果列表
id	int	文件主键
filename	str	文件名

下载公函模板

[POST] api/letter/download/

示例

请求：

代码块

```
1  {
2      "id" : 1,
3  }
```

响应：

对应的文件

请求参数

参数	类型	说明
id	int	文件主键

响应状态

参数	说明
error	存在问题（具体问题由包含的内容给出）
无	正常返回

响应数据

对应的文件

管理员删除模板

[POST] api/letter/delete-template/

示例

请求：

代码块

```
1  {
2      "id" : 1,
3  }
```

响应：

```
1  -- {
2      "message" : "操作成功",
3  }
```

请求参数

参数	类型	说明
id	int	文件主键

响应状态

返回提示	说明
message	操作成功
error	存在问题（具体问题由包含的内容给出）

支队长上传填写好的公函

[POST] api/letter/upload-letter/

示例

请求：
一个word或pdf文件

响应：

代码块

```
1  {
2      "message" : "上传成功",
3  }
```

请求参数

一个word或pdf文件

响应状态

返回提示	说明
message	操作成功
error	存在问题（具体问题由包含的内容给出）

管理员查询支队长填写好的公函

[GET] api/letter/query-letter/

示例

响应：

代码块

```
1  {
```

```
2  "count": 1,           // 总记录数
3  "next": null,         // 下一页链接（此处为 null 表示最后一页）
4  "previous": null,     // 上一页链接
5  "results": [          // 当前页数据（每页 20 条）
6    {
7      "id": 1,  // 主键
8      "filename" : "介绍信模板.docx", // 文件名
9      "sender" : { // 填写用户的信息
10         "id": 2,
11         "username": "wyk_back",
12         "email": "wangyukai@mails.tsinghua.edu.cn",
13         "student_id": null,
14         "phone_number": null,
15         "user_permission": "超级管理员"
16     },
17 },
18 ]
19 }
```

响应状态

参数	说明
error	存在问题（具体问题由包含的内容给出）
无	正常返回

响应数据

参数	类型	说明
count	int	数据数量
next	str	下一页的链接
previous	str	上一页的链接
results	list	结果列表
id	int	文件主键
filename	str	文件名
sender.id	int	用户主键
username	str	用户名
email	str	邮箱
student_id	str	学号
phone_number	str	手机号
user_permission	str	用户权限

管理员上传盖好章的公函

[POST] api/letter/upload-completed-letter/<int:id>/

例如：api/letter/upload-completed-letter/1/

由于未盖章的公函与管理员盖好章的公函是一对，所以前端需要通过url传入未盖章的公函模板的主键来告诉后端，请求体还是传入文件。

示例

请求：

一个word或pdf文件

响应：

代码块

```
1  {
2      "message" : "上传成功",
3  }
```

请求参数

一个word或pdf文件

响应状态

返回提示	说明
message	操作成功
error	存在问题（具体问题由包含的内容给出）

支队长查询公函状态

[GET] api/letter/query-status/

示例

响应：

代码块

```
1  {
2      "count": 2,
3      "next": null,
4      "previous": null,
5      "results": [
6          { // 第一种情况：已经有盖好章的回复
7              "letter": { // 填好了内容却没盖章的公函
8                  "id": 2,
9                  "filename": "数电书上习题.pdf"
10             },
11             "status": "已盖章",
12             "reply": { // 盖好章的公函
13                 "id": 5,
14                 "filename": "“数蜀拓新”实践支队_个人安全承诺书.docx"
15             }
16         },
17         { // 第二种情况：没有盖好章
18             "letter": { // 填好了内容却没盖章的公函
19                 "id": 6,
20                 "filename": "思想汇报.docx"
21             },
22             "status": "盖章中",
```

```
23         "reply": null // 没有盖章的
24     }
25 ]
26 }
```

响应状态

参数	说明
error	存在问题（具体问题由包含的内容给出）
无	正常返回

响应数据

参数	类型	说明
count	int	数据数量
next	str	下一页的链接
previous	str	上一页的链接
results	list	结果列表
letter.id	int	填写的公函主键
letter.filename	str	公函名
status	str	公函状态
reply.id	int	盖好章的公函主键
reply.filename	str	盖好章的公函名

实践日报填写

实践日报初始化

[POST] api/log/init/

示例

请求：

代码块

```
1  {
2      "id" : 1, // 支队主键
3      "research" : "前往四川成都调研“东数西算”有关企业，调研当前我国中西部地区大数据与超算发展情况", // 主要实践内容，不超过50字
4      "number" : 8, // 支队人数
5      "province" : "四川省", // 实践省份
6      "city" : "成都市", // 实践具体城市
7  }
```

响应：

```
代码块
1  {
2      "message" : "操作成功",
3  }
```

请求参数

参数	类型	说明
id	int	支队主键
research	str	调研内容
number	int	支队人数
province	str	实践省份
city	str	实践城市

响应状态

返回提示	说明
message	操作成功
error	存在问题（具体问题由包含的内容给出）

填写日报

[POST] api/log/write/

示例

请求：

```
代码块
1  {
2      "id" : 1, // 支队主键
3      "date" : "2025-05-13" / null, // 可有可无，如果没有则自动设置为当日
4      "content" : "今天干了啥", // 日报内容
5  }
```

响应：

```
代码块
1  {
2      "message" : "操作成功",
3  }
```

请求参数

参数	类型	说明
id	int	支队主键
date	str	日期

content	str	内容

响应状态

返回提示	说明
message	操作成功
error	存在问题（具体问题由包含的内容给出）

查询实践日报

[GET] api/log/query/

示例

响应：

代码块

```
1  {
2      "count": 1,
3      "next": null,
4      "previous": null,
5      "results": [
6          {
7              "detachment": {
8                  "id": 1,
9                  "name": "数蜀拓新",
10                 "research_content": "前往四川成都调研“东数西算”有关企业，调研当前我国中西部地区大数据与超算发展情况",
11                 "member_num": 8,
12                 "province": "四川省",
13                 "city": "成都市",
14                 "start_date": "2025-05-14",
15                 "end_date": "2025-05-17",
16                 "valid": true
17             },
18             "logs" : [
19                 {
20                     "date" : "2025-05-20",
21                     "content" : "今天没干啥",
22                 },
23                 {
24                     "date" : "2025-05-19",
25                     "content" : "今天啥也没干",
26                 }
27             ]
28         },
29     ]
30 }
```

响应状态

参数	说明
error	存在问题（具体问题由包含的内容给出）

无	正常返回
---	------

响应数据

参数	类型	说明
count	int	数据数量
next	str	下一页的链接
previous	str	上一页的链接
results	list	结果列表
id	int	支队主键
name	str	支队名
research_content	str	实践内容
member_num	int	支队人数
province	str	实践省份
city	str	实践城市
start_date	str	开始日期
end_date	str	结束日期
valid	bool	是否有效
logs.date	str	日报的日期
logs.content	str	日报的内容

投票评奖

创建问卷

[POST] api/vote/create/

示例

请求：

代码块

```
1  {
2      "title": "实践支队投票评奖问卷",
3      "permissions": ["超级管理员", "普通管理员"],
4      "questions": [
5          {
6              "question_idx": 1,
7              "question_text": "你认为哪个支队表现最好？ ",
8              "question_type": "single",
9              "options": ["支队 A", "支队 B", "支队 C"]
10         },
11         {
12             "question_idx": 2,
13             "question_text": "你认为支队的优点有哪些？ ",
14             "question_type": "multiple",
```

```
15         "options": ["组织有序", "成果显著", "团队合作好"]
16     },
17     {
18         "question_idx": 3,
19         "question_text": "请提出你的建议",
20         "question_type": "text"
21     },
22     {
23         "question_idx": 4,
24         "question_text": "打分",
25         "question_type": "score",
26         "min_score": 0,
27         "max_score": 10,
28         "step": 1
29     }
30 ]
31 }
```

响应：

代码块

```
1  {
2      "message": "成功创建问卷“
3  }
```

请求参数

参数	类型	说明
title	str	问卷标题
permissions	list	填写人群
questions	list	问题列表
question_idx	int	问题序号
question_text	str	问题内容
question_type	str	问题类型
options	list	选项
min_score	int	最小值
max_score	int	最大值
step	int	步长

响应状态

返回提示	说明
message	操作成功
error	存在问题（具体问题由包含的内容给出）

获取全部问卷

[GET] api/vote/get/?page=1

示例

响应：

代码块

```
1  {
2      "count": 2,
3      "next": null,
4      "previous": null,
5      "results": [
6          {
7              "id": 1,
8              "title": "实践支队投票评奖问卷",
9              "permissions": [
10                 "超级管理员",
11                 "普通管理员"
12             ],
13             "questions": [
14                 {
15                     "question_idx": 1,
16                     "question_text": "你认为哪个支队表现最好? ",
17                     "question_type": "single",
18                     "options": [
19                         "支队 A",
20                         "支队 B",
21                         "支队 C"
22                     ],
23                     "min_score": null,
24                     "max_score": null,
25                     "step": null
26                 },
27                 {
28                     "question_idx": 2,
29                     "question_text": "你认为支队的优点有哪些? ",
30                     "question_type": "multiple",
31                     "options": [
32                         "组织有序",
33                         "成果显著",
34                         "团队合作好"
35                     ],
36                     "min_score": null,
37                     "max_score": null,
38                     "step": null
39                 },
40                 {
41                     "question_idx": 3,
42                     "question_text": "请提出你的建议",
43                     "question_type": "text",
44                     "options": null,
45                     "min_score": null,
46                     "max_score": null,
47                     "step": null
48                 },
49                 {
50                     "question_idx": 4,
51                     "question_text": "打分",
```

```
52         "question_type": "score",
53         "options": null,
54         "min_score": 0,
55         "max_score": 10,
56         "step": 1.0
57     }
58 ]
59 ,is_published:true,
60 ,is_closed:true
61 },
62 {
63     "id": 2,
64     "title": "实践支队投票评奖问卷",
65     "permissions": [
66         "超级管理员",
67         "普通管理员"
68     ],
69     "questions": [
70         {
71             "question_idx": 1,
72             "question_text": "测试更新",
73             "question_type": "single",
74             "options": [
75                 "支队 A",
76                 "支队 B",
77                 "支队 C"
78             ],
79             "min_score": null,
80             "max_score": null,
81             "step": null
82         },
83         {
84             "question_idx": 2,
85             "question_text": "你认为支队的优点有哪些？ ",
86             "question_type": "multiple",
87             "options": [
88                 "组织有序",
```

响应状态

参数	说明
error	存在问题（具体问题由包含的内容给出）
无	正常返回

响应数据

参数	类型	说明
count	int	数据数量
next	str	下一页的链接
previous	str	上一页的链接
results	list	结果列表
id	str	问卷主键

title	str	问卷标题
permissions	list	填写人群
questions	list	问题列表
question_idx	int	问题序号
question_text	str	问题内容
question_type	str	问题类型
options	list	选项
min_score	int	最小值
max_score	int	最大值
step	int	步长
is_published	bool	是否发布
is_closed	bool	是否结束

发布问卷

[POST] api/vote/publish/

示例

请求：

代码块

```
1  {
2      "id": 1
3  }
```

响应：

代码块

```
1  {
2      "message": "成功发布问卷"
3  }
```

请求参数

参数	类型	说明
id	int	问卷主键

响应状态

返回提示	说明
message	操作成功
error	存在问题（具体问题由包含的内容给出）

截止收集问卷

[POST] api/vote/close/

示例

请求：

代码块

```
1  {
2      "id": 1
3  }
```

响应：

代码块

```
1  {
2      "message": "成功停止收集问卷"
3  }
```

请求参数

参数	类型	说明
id	int	问卷主键

响应状态

返回提示	说明
message	操作成功
error	存在问题（具体问题由包含的内容给出）

用户提交问卷

[POST] api/vote/submit/

示例

请求：

代码块

```
1  {
2      "id": 1,
3      "answers": [
4          {
5              "question_idx": 1,
6              "answer": "选项1"
7          },
8          {
9              "question_idx": 2,
10             "answer": ["选项1","选项2"]
11         },
12         {
13             "question_idx": 3,
```

```
14         "answer": "希望大家继续加油"
15     },
16     {
17         "question_idx": 4,
18         "answer": 100
19     }
20 ]
21 }
```

响应：

代码块

```
1  {
2      "message": "成功提交问卷"
3  }
```

请求参数

参数	类型	说明
id	int	问卷主键
answers	list	答案列表
question_idx	int	问题序号
answer	str/list/int	回答

响应状态

返回提示	说明
message	操作成功
error	存在问题（具体问题由包含的内容给出）

查询问卷结果

[POST] api/vote/result/

示例

请求：

代码块

```
1  {
2      "id": 1
3  }
```

响应：

代码块

```
1  {
```



```
2      "results": [
3          {
4              "user_id": 1,
5              "answers": [
6                  {
7                      "question_idx": 1,
8                      "answer": [1]
9                  },
10                 {
11                     "question_idx": 2,
12                     "answer": [1, 2]
13                 },
14                 {
15                     "question_idx": 3,
16                     "answer": "希望大家继续加油"
17                 }
18             ]
19         }
20     ]
```

请求参数

参数	类型	说明
id	int	问卷主键

响应状态

参数	说明
error	存在问题（具体问题由包含的内容给出）
无	正常返回

响应数据

参数	类型	说明
results	list	结果列表
user_id	int	用户主键
answers	list	答案列表
question_idx	int	问题序号
answer	str/list/int	回答

删除问卷

[POST] api/vote/delete/

示例

请求：

```
1  {
2      "id": 1
3  }
```

响应：

代码块

```
1  {
2      "message": "成功删除问卷"
3  }
```

请求参数

参数	类型	说明
id	int	问卷主键

响应状态

返回提示	说明
message	操作成功
error	存在问题（具体问题由包含的内容给出）