

接口文档

方式: http

域名: api.inaction.fun

若无特殊说明, 请求和返回数据都是 json 格式

注册

url: /register

方法: post

参数:

json格式

key	类型	说明
account	string	账号
password	string	密码
name	string	名称

返回数据:

状态码	数据	说明
200	{"msg":"ok"}	成功
409	{"msg":"账号已存在"}	账号已存在

修改用户信息

url: /update/user/data

方法： post

请求参数：

key	类型	说明
password	string	密码
name	string	名称

返回数据：

状态码	数据	说明
200	{"msg":"ok"}	成功

登录

url： /login

方法： post

参数：

json格式

key	类型	说明
account	string	账号
password	string	密码

返回数据：

状态码	数据	说明
200	{"msg":"ok"}	成功
404	{"msg":"账号不存在"}	账号不存在
409	{"msg":"密码错误"}	密码错误

登出

url: /logout

方法: get

创建项目

url: /new/project

方法: post

请求参数:

json格式

key	类型	说明
name	string	项目名
members	json数组	项目成员（不包括组长）

members

key	类型	说明
account	string	成员的账号
permission	int	权限，0代表只读，1代表可读可改

返回数据:

状态码	数据	说明
200	{"projectId":"项目id"}	成功，并返回项目id
404	{ "projectId":"项目id", "notFound":["不存在的成员账号1","账号2"] }	某些成员账号不存在，此时创建项目成功，但是某些成员未加入项目

获取项目基本信息

url: /find/project?id=xxxx

方法: get

url参数: 项目的id

返回数据:

```
1  {
2      "name" : "String", // 项目名
3      "createTime" : "Long", // 创建时间
4      "creatorAccount" : "String", // 创建者账号
5      "creatorName": "String", // 创建者的名称
6      "members" : [
7          {
8              "userId" : "String", // 成员的ObjectId
9              "userAccount" : "String", // 成员的账号
10             "userName" : "String", // 成员的名称
11             "permission" : "Int" // 权限, 0只读, 1可读可写
12         }
13     ]
14 }
```

获取项目API信息

url: /find/project/apis?id=xxxx

方法: get

url参数: 项目的id

返回数据:

```
1  [
2      {
3          "groupName": "String", // API分组名称
4          "apiIds": [
5              {
6                  "apiId": "String", // api的id
7                  "name": "String", // api的名称
8              }
9          ]
10     }
11 ]
```

添加项目成员

url: /new/project/member

方法: post

请求参数:

key	类型	说明
projectId	string	项目id
account	string	成员的账号
permission	int	权限, 0代表只读, 1代表可读可改

返回数据:

状态码	数据	说明
200	{"msg":"ok"}	成功
404	{"msg":"用户账号或项目不存在"}	成员账号或项目不存在
409	{"msg":"不能重复加入"}	该用户已是项目成员

删除项目成员

url: /delete/project/member

方法: post

请求参数:

key	类型	说明
projectId	string	项目id
account	string	成员的账号

返回数据:

状态码	数据	说明
200	{"msg":"ok"}	成功
404	{"msg":"用户账号或项目不存在，或用户不是项目成员"}	成员账号或项目不存在

修改项目成员权限

url: /update/member/permission

方法: post

请求参数:

key	类型	说明
projectId	string	项目id
account	string	成员的账号
permission	int	权限，0代表只读，1代表可读可改

返回数据:

状态码	数据	说明
200	{"msg":"ok"}	成功
405	{"msg":"client error"}	失败，可能由发送请求者非项目组长或成员不属于项目等原因造成

获取用户信息

url: /user/data

方法: get

返回参数:

```
1  {
2    "name": "用户名称",
3    "account": "用户账号",
```

```
4     "password": "用户密码",
5     "project": [
6         {
7             "id": "项目id",
8             "name": "项目名",
9             "createTime": "项目创建时间",
10            "leaderAccount": "组长账号",
11            "leaderName": "组长名称",
12            "members": [
13                {
14                    "account": "成员的账号",
15                    "name": "成员的昵称",
16                    "permission": "权限（1或2）"
17                }
18            ]
19        }
20    ]
21 }
```

创建分组

url: /new/project/api_group

方法: post

请求参数:

key	类型	说明
projectId	string	项目id
groupName	string	分组名

返回数据:

状态码	数据	说明
200	{"msg":"ok"}	成功
404	{"msg":"not found"}	失败, 项目id不存在
409	{"msg":"该分组名已存在"}	失败, 该分组名已存在

创建接口

url: /new/api

方法: post

请求参数:

每个字段都必须有, 如果数据为空填入null

```
1  {
2      "projectId" : "String", // 项目id
3      "name" : "String", // 接口名称
4      "protocol" : "Int", // 协议
5      "url" : "String", // 接口的url地址
6      "group" : "String", // 接口的组名
7      "status" : "Int", // 接口的状态
8      "explain" : "String", // 接口的说明信息
9      "requestMethod" : "Int", // 请求方法
10     "urlParam" : [
11         {
12             "paramKey" : "String", // 参数名
13             "paramExplain" : "String" // 参数说明
14         }
15     ],
16     "requestHeader" : [ // 请求头
17         {
18             "headerName" : "String", // 请求头名
19             "headerValue" : "String", // 值
20             "explain" : "String" // 说明
21         }
22     ],
23     "requestParamType" : "Int", // 请求参数类型
24     "requestParamJsonType" : "Int", // 请求参数的JSON类型 (Object还是
Array)
25     "requestParam" : [ // 接口的请求参数
26         {
27             "paramKey" : "String", // 参数名
28             "type" : "String", // 类型
29             "explain" : "String", // 说明
30             "childList" : [
31                 requestParam
32             ]
33         }
34     ],
35     "requestRaw" : "String", // 请求参数, raw类型
36     "requestExplain" : "String", // 请求数据的额外说明信息
37     "responseData" : [ // 返回数据, 用数组是因为可能有多种情况
38         {
39             "responseHeader" : [
40                 {
41                     "headerName" : "String", // 返回头名
42                     "headerValue" : "String", // 值
43                     "explain" : "String" // 说明
```



```

44         }
45     ],
46     "responseParamType" : "Int", // 返回参数类型
47     "responseParamJsonType" : "Int", // 请求参数的JSON类型
48     "responseParam" : [
49         {
50             "paramKey" : "String", // 参数名
51             "type" : "String", // 类型
52             "explain" : "String", // 说明
53             "childList" : [
54                 responseParam
55             ]
56         }
57     ],
58     "responseRaw" : "String", // 返回数据, raw类型
59     "responseExplain" : "String" // 返回数据的额外说明信息
60 }
61 ]
62 }

```

- **procotol (协议)**

值	说明
1	http
2	https

- **status (接口的状态)**

值	说明
1	已发布
2	开发中
3	废弃

- **requestMethod (请求方法)**

值	说明
1	get
2	post
3	put
4	delete
5	head
6	options
7	patch

- requestParamType/responseParamType (请求/返回 参数类型)

值	说明
1	Form-data
2	json
3	raw
4	其他

- requestParamJsonType/responseParamType (请求/返回 的json数据的根类型)

值	说明
1	Object
2	Array

返回参数:

状态码	数据	说明
200	{"msg": "ok" , "apild" : "api的id" }	创建成功
403	{"msg": "no permission"}	无权限
406	{"msg" : "错误原因"}	创建失败

删除接口

url: /delete/api?id=xxxxx

方法: get

url参数: api的id

返回参数:

状态码	数据	说明
200	{"msg": "ok"}	删除成功
404	{"msg": "api id not found"}	api的id不存在
403	{"msg": "no permission"}	无权限

查询接口信息

url: /find/api?id=xxxxxx

方法: get

url参数: api的id

返回参数:

```
1  {
2      "projectId": "String", // 所属的项目id
3      "name" : "String", // 接口名称
4      "protocol" : "Int", // 协议, 0为http, 1为https
5      "url" : "String", // 接口的url地址
6      "group" : "String", // 接口的组名
7      "status" : "Int", // 接口的状态
8      "explain" : "String", // 接口的说明信息
9      "createTime" : "Long", // 接口的创建时间
10     "createUser" : "UserId", // 接口的创建者
11     "updateTime" : "Long", // 接口的最后更新时间
12     "updateUser" : "UserId", // 接口的最后更新人
13     "updateInfo" : "String", // 接口的更新说明
14     "requestMethod" : "Int", // 请求方法
15     "urlParam" : [
16         {
17             "paramKey" : "String", // 参数名
18             "paramExplain" : "String" // 参数说明
19         }
20     ],
21     "requestHeader" : [ // 请求头
22         {
23             "headerName" : "String", // 请求头名
24             "headerValue" : "String", // 值
25             "explain" : "String" // 说明
26         }
27     ],
28     "requestParamType" : "Int", // 请求参数类型
29     "requestParamJsonType" : "Int", // 请求参数的JSON类型 (Object还是
Array)
30     "requestParam" : [ // 接口的请求参数
31         {
32             "paramKey" : "String", // 参数名
33             "type" : "String", // 类型
34             "explain" : "String", // 说明
35             "childList" : [
36                 requestParam
37             ]
38         }
39     ],
```

```

40     "requestRaw" : "String", // 请求参数, raw类型
41     "requestExplain" : "String", // 请求数据的额外说明信息
42     "responseData" : [ // 返回数据, 用数组是因为可能有多种情况
43         {
44             "responseHeader" : [
45                 {
46                     "headerName" : "String", // 返回头名
47                     "headerValue" : "String", // 值
48                     "explain" : "String" // 说明
49                 }
50             ],
51             "responseParamType" : "Int", // 返回参数类型
52             "requestParamJsonType" : "Int", // 请求参数的JSON类型
53             "responseParam" : [
54                 {
55                     "paramKey" : "String", // 参数名
56                     "type" : "String", // 类型
57                     "explain" : "String", // 说明
58                     "childList" : [
59                         responseParam
60                     ]
61                 }
62             ],
63             "responseRaw" : "String",
64             "responseExplain" : "String" // 返回数据的额外说明信息
65         }
66     ]
67 }

```

- **procotol (协议)**

值	说明
1	http
2	https

- **status (接口的状态)**

值	说明
1	已发布
2	开发中
3	废弃

- **requestMethod (请求方法)**

值	说明
1	get
2	post
3	put
4	delete
5	head
6	options
7	patch

• requestParamType/responseParamType (请求/返回 参数类型)

值	说明
1	Form-data
2	json
3	raw
4	其他

• requestParamJsonType/responseParamType (请求/返回 的json数据的根类型)

值	说明
1	Object
2	Array

修改接口信息

url: /update/api?id=xxxx&&info=xxxxx

方法: post

url参数: id为api的id, info是接口的更新说明

请求参数:

与创建接口的格式相同

```
1  {
2      "projectId" : "String", // 项目id
```

```

3      "name" : "String", // 接口名称
4      "protocol" : "Int", // 协议
5      "url" : "String", // 接口的url地址
6      "group" : "String", // 接口的组名
7      "status" : "Int", // 接口的状态
8      "explain" : "String", // 接口的说明信息
9      "requestMethod" : "Int", // 请求方法
10     "urlParam" : [
11         {
12             "paramKey" : "String", // 参数名
13             "paramExplain" : "String" // 参数说明
14         }
15     ],
16     "requestHeader" : [ // 请求头
17         {
18             "headerName" : "String", // 请求头名
19             "headerValue" : "String", // 值
20             "explain" : "String" // 说明
21         }
22     ],
23     "requestParamType" : "Int", // 请求参数类型
24     "requestParamJsonType" : "Int", // 请求参数的JSON类型 (Object还是
Array)
25     "requestParam" : [ // 接口的请求参数
26         {
27             "paramKey" : "String", // 参数名
28             "type" : "String", // 类型
29             "explain" : "String", // 说明
30             "childList" : [
31                 requestParam
32             ]
33         }
34     ],
35     "requestRaw" : "String", // 请求参数, raw类型
36     "requestExplain" : "String", // 请求数据的额外说明信息
37     "responseData" : [ // 返回数据, 用数组是因为可能有多种情况
38         {
39             "responseHeader" : [
40                 {
41                     "headerName" : "String", // 返回头名
42                     "headerValue" : "String", // 值
43                     "explain" : "String" // 说明
44                 }
45             ],
46             "responseParamType" : "Int", // 返回参数类型
47             "responseParamJsonType" : "Int", // 请求参数的JSON类型
48             "responseParam" : [
49                 {
50                     "paramKey" : "String", // 参数名
51                     "type" : "String", // 类型
52                     "explain" : "String", // 说明
53                     "childList" : [
54                         responseParam
55                     ]
56                 }

```

```
57         ],
58         "responseRaw" : "String", // 返回数据, raw类型
59         "responseExplain" : "String" // 返回数据的额外说明信息
60     }
61 ]
62 }
```

获取接口历史信息

url: /find/api/history?id=xxxxx&&type=x

方法: get

url参数:

key	说明
id	api的id
type	0 代表获取简略信息（不包括api的具体数据） 1 代表获取详细信息（包括api的具体数据）

返回数据:

- 当 type = 0 时

```
1  {
2      "_id": "ObjectId",
3      "apiId": "string", // 对应的api的id
4      "history": [
5          {
6              "updateTime": "Long", // 更新时间
7              "updateUserName": "String", // 更新人的名称
8              "updateInfo": "String", // 更新说明
9          }
10     ]
11 }
```

- 当 type = 1 时

```
1  {
2      "_id": "ObjectId",
3      "apiId": "string", // 对应的api的id
4      "history": [
5          {
6              "updateTime": "Long", // 更新时间
7              "updateUserName": "String", // 更新人的名称
8              "updateInfo": "String", // 更新说明
9              "api": "ApiObject" // api数据, 格式完全对应接口表
10         }
11     ]
12 }
```