

00A0 2203∃ 2200∀ 2286⊆ 2713x 27FA⇔ 221A√ 221B³/ 2295⊕ 2297⊗ UTF8gbsn

Studio Documentation

发布 *1.0.0*

Studio team

2021 年 01 月 19 日

Contents

1	开发文档	1
1.1	API 文档	1

1.1 API 文档

1.1.1 资源组

说明

- 获取流程调试日志

1、资源组查询

请求方式： GET

请求地址： /api/control/resource

Content-Type:

application/x-www-form-urlencoded —— 表示通过表单方式提交

查询参数:

参数	数据类型	是否必须	说明
id	string	否	pk(无需输入)
name	string	否	名称
cuser	string	否	创建用户
description	string	否	描述
is_default	int	否	是否默认

输入参数:

参数	数据类型	是否必须	说明
id	string	否	pk(无需输入)
name	string	是	名称
ctime	string	是	创建时间
cuser	string	否	创建用户
description	string	否	描述
is_default	int	否	是否默认

输出参数:

参数	数据类型	是否必须	说明
id	string	是	pk(无需输入)
name	string	是	名称
cuser	string	是	创建用户
description	string	是	描述
is_default	int	否	是否默认

日志查询请求示例:

- Content-Type: 'application/json'

```
{
  "id": "",
  "name": "",
  "ctime": "",
  "cuser": "",
  "description": "",
  "is_default": ""
}
```

- Content-Type: 'application/x-www-form-urlencoded'

```
curl http://ip:port/api/management/resource?id=&name=&cuser&description=&is_default=
```

日志查询返回数据示例:

- Content-Type: 'application/json'

```
{
  "count": 4,
  "next": null,
  "previous": null,
  "results": [
    {
      "id": 8,
      "name": "基础运维部",
      "cuser": "xiaofeng",
      "ctime": 1602638786,
      "description": ""
    },
    {
      "id": 6,
      "name": "测试",
      "cuser": "admin",

```

(下页继续)

(续上页)

```
        "ctime": 1597368265,
        "description": "测试阶段使用"
    },
    {
        "id": 5,
        "name": "财务部",
        "cuser": "admin",
        "ctime": 1596519868,
        "description": ""
    },
    {
        "id": 1,
        "name": "Default",
        "cuser": "admin",
        "ctime": null,
        "description": null
    }
  ]
}
```

日志查询错误返回示例:

```
{
  "detail": "资源组获取失败"
}
```

1.1.2 机器人

说明

- 机器人管理

1、机器人列表

请求方式: GET

请求地址: /api/control/robot

Content-Type:

application/x-www-form-urlencoded —— 表示通过表单方式提交

查询参数:

参数	数据类型	是否必须	说明
id	string	否	pk(无需输入)
name	string	否	应用名称, String
state	string	否	状态: online-在线; offline-离线; busy-忙碌
resource	string	否	资源组 ID, String
activate_state	String	否	激活状态, String
atime_from	long	否	激活时间 (from), long
atime_to	long	否	激活时间 (to), long
last_time_from	long	否	最后通讯时间 (from), long
last_time_to	long	否	最后通讯时间 (to), long

输入参数:

参数	数据类型	是否必须	说明
id	string	否	pk(无需输入)
name	string	否	应用名称, String
state	string	否	状态: unactivated-未激活; free-空闲; busy-忙碌; unreachable-无响应
resource	string	否	资源组 ID, String
atime	int	否	激活时间
remark	string	否	备注, Text
last_time	int	否	最后通讯时间, int
ctime	int	否	创建时间, int
version	String	否	版本, String
ip	String	否	IP, String
credential	String	否	凭证 ID, String
computer_name	String	否	计算机名称, string

输出参数:

参数	数据类型	是否必须	说明
id	string	否	pk(无需输入)
name	string	否	应用名称, String
state	string	否	状态: unactivated-未激活; free-空闲; busy-忙碌; unreachable-无响应
resource	string	否	资源组 ID, String
atime	int	否	激活时间
remark	string	否	备注, Text
last_time	int	否	最后通讯时间, int
ctime	int	否	创建时间, int
version	String	否	版本, String
ip	String	否	IP, String
credential	String	否	凭证 ID, String
computer_name	String	否	计算机名称, string

机器人列表请求示例:

- Content-Type: 'application/json'

```
{
  "id": "",
  "name": "",
  "state": "",
  "resource": "",
  "remark": "",
  "atime": "",
  "last_time": "",
  "ctime": "",
  "version": "",
  "ip": "",
  "credential": "",
  "computer_name": ""
}
```

- Content-Type: 'application/x-www-form-urlencoded'

```
curl http://ip:port/api/robot/robot?id=&name=&state=&resource=&remark=&atime=&last_
↪time=&ctime=&version=&ip=&credential=&computer_name=
```

机器人列表返回数据示例:

- Content-Type: 'application/json'

```
{
  "count": 14,
  "next": null,
  "previous": null,
  "results": [
    {
      "id": "d206cb06-1db2-11eb-bab5-0001228265cb",
      "resource": 1,
      "name": "robotx",
      "version": "1.0",
      "activate_state": "unactivated",
      "state": "offline",
      "ip": "192.168.10.28",
      "credential": 45,
      "last_time": 1606288894,
      "ctime": 1604393942,
      "atime": 1606288837,
      "remark": null,
      "state_name": "离线",
      "activate_state_name": "未激活",
      "credential_name": "jackson_windows",
      "connection_id": null,
      "template_ids": [
        145,
        148
      ],
      "template_list": [
        {
          "id": 145,
          "name": "查询机票",
          "project": 87
        },
        {
          "id": 148,
```

(下页继续)

(续上页)

```

        "name": "点击测试",
        "project": 89
    }
],
"computer_name": ""
},
{
    "id": "ce8699dc-2271-11eb-9d93-0000442dcdf9",
    "resource": 1,
    "name": "大主机",
    "version": "1.0",
    "activate_state": "activated",
    "state": "offline",
    "ip": "192.168.10.97",
    "credential": 27,
    "last_time": 1605268199,
    "ctime": 1604915775,
    "atime": 1604915801,
    "remark": null,
    "state_name": "离线",
    "activate_state_name": "已激活",
    "credential_name": "windows 机器",
    "connection_id": 23,
    "template_ids": [
        148
    ],
    "template_list": [
        {
            "id": 148,
            "name": "点击测试",
            "project": 89
        }
    ],
    "computer_name": ""
},
{
    "id": "c439f0c8-2582-11eb-9ddd-0001eb7e19ed",
    "resource": 1,
    "name": "飞力达 RPA",
    "version": "1.0",
    "activate_state": "activated",
    "state": "offline",
    "ip": "192.168.10.227",
    "credential": 27,
    "last_time": 1606293889,
    "ctime": 1605252913,
    "atime": 1605252922,
    "remark": null,
    "state_name": "离线",
    "activate_state_name": "已激活",
    "credential_name": "windows 机器",
    "connection_id": 25,
    "template_ids": [
        155,
        160,
        162,
    ]
}

```

(下页继续)

(续上页)

```

        163,
        165,
        179,
        186
    ],
    "template_list": [
        {
            "id": 155,
            "name": "宁波港",
            "project": 83
        },
        {
            "id": 160,
            "name": "太仓港 1",
            "project": 83
        },
        {
            "id": 162,
            "name": "太仓港 2",
            "project": 83
        },
        {
            "id": 163,
            "name": "上海港",
            "project": 83
        },
        {
            "id": 165,
            "name": "太仓港 3",
            "project": 83
        },
        {
            "id": 179,
            "name": "飞力达 excel 操作",
            "project": 83
        },
        {
            "id": 186,
            "name": "51job 简历上传",
            "project": 94
        }
    ],
    "computer_name": ""
},
{
    "id": "b4d5cbb8-0d24-11eb-81e6-0000b0adc4d1",
    "resource": 1,
    "name": "JAVIS",
    "version": "1.0",
    "activate_state": "activated",
    "state": "offline",
    "ip": "172.16.200.164",
    "credential": 33,
    "last_time": 1602583224,
    "ctime": 1602573686,
    "atime": 1602574462,

```

(下页继续)

(续上页)

```

    "remark": null,
    "state_name": "离线",
    "activate_state_name": "已激活",
    "credential_name": "Jackson 机器登录凭据",
    "connection_id": null,
    "template_ids": [],
    "template_list": [],
    "computer_name": ""
  },
  {
    "id": "b413da38-2e2f-11eb-af83-000027b6bfd1",
    "resource": 1,
    "name": "tinkpad",
    "version": "1.0",
    "activate_state": "activated",
    "state": "online",
    "ip": "192.168.10.132",
    "credential": 50,
    "last_time": 1606906669,
    "ctime": 1606206798,
    "atime": 1606206857,
    "remark": null,
    "state_name": "在线",
    "activate_state_name": "已激活",
    "credential_name": "tinkpad 登录",
    "connection_id": 26,
    "template_ids": [
      163,
      188
    ],
    "template_list": [
      {
        "id": 163,
        "name": "上海港",
        "project": 83
      },
      {
        "id": 188,
        "name": "舒尔 rpa 整合流程",
        "project": 96
      }
    ],
    "computer_name": ""
  },
  {
    "id": "a7f154a8-2eee-11eb-bba1-0001f11db06e",
    "resource": 1,
    "name": "win7",
    "version": "1.0",
    "activate_state": "activated",
    "state": "offline",
    "ip": "192.168.10.28",
    "credential": 50,
    "last_time": 1606289224,
    "ctime": 1606288811,
    "atime": 1606288903,

```

(下页继续)

(续上页)

```

    "remark": null,
    "state_name": "离线",
    "activate_state_name": "已激活",
    "credential_name": "tinkpad 登录",
    "connection_id": 28,
    "template_ids": [
        188
    ],
    "template_list": [
        {
            "id": 188,
            "name": "舒尔 rpa 整合流程",
            "project": 96
        }
    ],
    "computer_name": ""
},
{
    "id": "a0194f86-2319-11eb-8bce-00010daea7a4",
    "resource": 1,
    "name": "泛汐机器人",
    "version": "1.0",
    "activate_state": "activated",
    "state": "offline",
    "ip": "172.16.0.181",
    "credential": 27,
    "last_time": 1605233217,
    "ctime": 1604987853,
    "atime": 1604987868,
    "remark": null,
    "state_name": "离线",
    "activate_state_name": "已激活",
    "credential_name": "windows 机器",
    "connection_id": null,
    "template_ids": [
        148
    ],
    "template_list": [
        {
            "id": 148,
            "name": "点击测试",
            "project": 89
        }
    ],
    "computer_name": ""
},
{
    "id": "8e62dde6-2d3c-11eb-be1d-0001ea721897",
    "resource": 1,
    "name": "test1",
    "version": "",
    "activate_state": "unactivated",
    "state": "",
    "ip": "",
    "credential": 42,
    "last_time": null,

```

(下页继续)

(续上页)

```

    "ctime": 1606102367,
    "atime": null,
    "remark": null,
    "state_name": "",
    "activate_state_name": "未激活",
    "credential_name": "测试凭据",
    "connection_id": null,
    "template_ids": [
        164
    ],
    "template_list": [
        {
            "id": 164,
            "name": "ceshiliucheng",
            "project": 89
        }
    ],
    "computer_name": ""
},
{
    "id": "88a7af70-1cd1-11eb-abf4-0001bd1fe9d6",
    "resource": 1,
    "name": "jackson",
    "version": "1.0",
    "activate_state": "activated",
    "state": "offline",
    "ip": "192.168.10.143",
    "credential": 45,
    "last_time": 1605859354,
    "ctime": 1604297183,
    "atime": 1604297219,
    "remark": null,
    "state_name": "离线",
    "activate_state_name": "已激活",
    "credential_name": "jackson_windows",
    "connection_id": 15,
    "template_ids": [
        144
    ],
    "template_list": [
        {
            "id": 144,
            "name": "网银转账",
            "project": 86
        }
    ],
    "computer_name": ""
},
{
    "id": "766d357a-1fd1-11eb-80ea-0001e1938a54",
    "resource": 1,
    "name": "David_ 测试",
    "version": "1.1",
    "activate_state": "activated",
    "state": "offline",
    "ip": "192.168.10.130",

```

(下页继续)

(续上页)

```

        "credential": 42,
        "last_time": 1606362116,
        "ctime": 1604627006,
        "atime": 1606806717,
        "remark": null,
        "state_name": "离线",
        "activate_state_name": "已激活",
        "credential_name": "测试凭据",
        "connection_id": 30,
        "template_ids": [
            139,
            187
        ],
        "template_list": [
            {
                "id": 139,
                "name": "测试网站点击",
                "project": 82
            },
            {
                "id": 187,
                "name": "david 测试 jipr_text",
                "project": 95
            }
        ],
        "computer_name": ""
    },
    {
        "id": "6a8962de-2556-11eb-94a5-000048fdab24",
        "resource": 1,
        "name": "泛汐机器人 RPA",
        "version": "1.0",
        "activate_state": "activated",
        "state": "online",
        "ip": "172.16.0.181",
        "credential": 27,
        "last_time": 1606906670,
        "ctime": 1605233864,
        "atime": 1605233912,
        "remark": null,
        "state_name": "在线",
        "activate_state_name": "已激活",
        "credential_name": "windows 机器",
        "connection_id": null,
        "template_ids": [
            155,
            160,
            162,
            163,
            165,
            179
        ],
        "template_list": [
            {
                "id": 155,
                "name": "宁波港",

```

(下页继续)

(续上页)

```

        "project": 83
      },
      {
        "id": 160,
        "name": "太仓港 1",
        "project": 83
      },
      {
        "id": 162,
        "name": "太仓港 2",
        "project": 83
      },
      {
        "id": 163,
        "name": "上海港",
        "project": 83
      },
      {
        "id": 165,
        "name": "太仓港 3",
        "project": 83
      },
      {
        "id": 179,
        "name": "飞力达 excel 操作",
        "project": 83
      }
    ],
    "computer_name": ""
  },
  {
    "id": "60ad6ea8-1fe1-11eb-b50d-00010a56bc23",
    "resource": 1,
    "name": "robot-test",
    "version": "1.0",
    "activate_state": "activated",
    "state": "offline",
    "ip": "192.168.10.17",
    "credential": 27,
    "last_time": 1604644949,
    "ctime": 1604633841,
    "atime": 1604644893,
    "remark": null,
    "state_name": "离线",
    "activate_state_name": "已激活",
    "credential_name": "windows 机器",
    "connection_id": 19,
    "template_ids": [
      150
    ],
    "template_list": [
      {
        "id": 150,
        "name": "测试无分支流程-复制",
        "project": null
      }
    ]
  }

```

(下页继续)

(续上页)

```

    ],
    "computer_name": ""
  },
  {
    "id": "4da91052-2f99-11eb-9faf-0001f3cf97e9",
    "resource": 1,
    "name": "飞力达 POC 机器人",
    "version": "1.0",
    "activate_state": "activated",
    "state": "offline",
    "ip": "192.168.253.1",
    "credential": 50,
    "last_time": 1606437906,
    "ctime": 1606362104,
    "atime": 1606362164,
    "remark": null,
    "state_name": "离线",
    "activate_state_name": "已激活",
    "credential_name": "tinkpad 登录",
    "connection_id": 29,
    "template_ids": [
      163
    ],
    "template_list": [
      {
        "id": 163,
        "name": "上海港",
        "project": 83
      }
    ],
    "computer_name": ""
  },
  {
    "id": "2d4e28d2-fe37-11ea-ac0f-000151a558a6",
    "resource": 1,
    "name": "181",
    "version": "1.0",
    "activate_state": "activated",
    "state": "offline",
    "ip": "172.16.0.181",
    "credential": 27,
    "last_time": 1605233775,
    "ctime": 1600932352,
    "atime": 1600992163,
    "remark": null,
    "state_name": "离线",
    "activate_state_name": "已激活",
    "credential_name": "windows 机器",
    "connection_id": 11,
    "template_ids": [
      132
    ],
    "template_list": [
      {
        "id": 132,
        "name": "测试",

```

(下页继续)

(续上页)

```

        "project": 56
      }
    ],
    "computer_name": ""
  }
]
}

```

机器人列表错误返回示例:

```

{
  "detail": "资源组获取失败"
}

```

1.1.3 主机清单

说明

- 获取主机清单

1、主机清单查询

请求方式: GET

请求地址: /api/control/inventory/

Content-Type:

application/x-www-form-urlencoded —— 表示通过表单方式提交
application/json —— 表示传入数据为 json 格式字符串

GET 查询参数:

参数	数据类型	是否必须	说明
id	int	是	inventory id

返回数据示例:

- Content-Type: 'application/json'

```

{
  "count": 57,
  "next": "http://opsgnat-tst.opsgnat.com/api/project/inventory?limit=20&offset=20",
  "previous": null,
  "results": [
    {
      "id": 559,
      "resource": 1,
      "name": "xxxx",
      "description": "",
      "variables": "",
      "cuser": "admin",

```

(下页继续)

(续上页)

```

        "credential_id": null,
        "status_health": 0,
        "status_running": 0,
        "status_exception": 0,
        "status_disabled": 0,
        "group_counts": 0,
        "host_counts": 0,
        "has_host_source": false,
        "credential": null,
        "source": "internal",
        "source_name": "手工录入"
    },
    {
        "id": 156,
        "resource": 6,
        "name": "测试 009",
        "description": "",
        "variables": "",
        "cuser": "xiaofeng",
        "credential_id": null,
        "status_health": 0,
        "status_running": 0,
        "status_exception": 0,
        "status_disabled": 0,
        "group_counts": 3,
        "host_counts": 2,
        "has_host_source": false,
        "credential": null,
        "source": "internal",
        "source_name": "手工录入"
    },
    {
        "id": 142,
        "resource": 1,
        "name": "本机 (localhost)",
        "description": "",
        "variables": "",
        "cuser": "admin",
        "credential_id": 34,
        "status_health": 0,
        "status_running": 0,
        "status_exception": 1,
        "status_disabled": 0,
        "group_counts": 0,
        "host_counts": 1,
        "has_host_source": false,
        "credential_name": "101gitlab 凭据",
        "credential": 34,
        "source": "internal",
        "source_name": "手工录入"
    },
]
}

```

2、获取单个主机清单 API

请求方式： GET （查询）

请求地址： /api/process/process/559/

请求地址中 559 为主机清单的 id

返回数据例子：

```
{
  "id": 559,
  "resource": 1,
  "name": "xxxx",
  "description": "",
  "variables": "",
  "cuser": "admin",
  "credential_id": null,
  "status_health": 0,
  "status_running": 0,
  "status_exception": 0,
  "status_disabled": 0,
  "group_counts": 0,
  "host_counts": 0,
  "has_host_source": false,
  "credential": null,
  "source": "internal",
  "source_name": "手工录入"
}
```

1.1.4 Process Document

流程

说明

- 流程增删该查

1、流程列表、新增流程、批量删除流程 API

请求方式： GET （查询） POST （新增） DELETE （批量删除）

请求地址： /api/process/process/

Content-Type:

新增数据的时候需要指定 Content-Type，以下对 Content-Type 进行说明：
application/x-www-form-urlencoded —— 表示通过表单方式提交
application/json —— 表示传入数据为 json 格式字符串

查询参数：

参数	数据类型	是否必须	说明
name	string	是	项目名称

输入参数 (新增):

参数	数据类型	是否必须	说明
name	string	是	项目名称
project	string	是	项目地址
mark	string	否	备注

输出参数:

参数	数据类型	是否必须	说明
name	string	是	项目名称
project	string	是	项目地址
mark	string	否	备注

批量删除参数:

参数	数据类型	是否必须	说明
pk	string	与 pk[] 不能都为空	主键, 多个主键用半角逗号隔开。通过 http body 传入
pk[]	array	与 pk 不能都为空	主键数组。通过 http body 传入

排序:

参数	数据类型	是否必须	说明
ordering	string	否	一般默认按 id 倒叙 <ul style="list-style-type: none"> ordering=id 表示按 id 排序 ordering=-id 表示按 id 倒叙 多个字段排序用半角逗号分隔

GET 返回数据例子:

```
{
  "count": 9,
  "next": null,
  "previous": null,
  "results": [
    {
      "id": 1001,
      "name": "测试",
      "cuser": 18,
      "user": "san zhang",
      "ctime": 1561688923,
      "utime": 1561689923,
      "project": "测试",
      "mark": "新增测试流程",
    },
    {
      "id": 1002,
      "name": "测试",
    }
  ]
}
```

(下页继续)

(续上页)

```
        "cuser": 18,
        "user": "si li",
        "ctime": 1561688924,
        "utime": 1561689924,
        "project": "测试 2",
        "mark": "新增测试流程 2",
    }
    {
        "id": 1003,
        "name": "测试",
        "cuser": 18,
        "user": "wu wang",
        "ctime": 1561688925,
        "utime": 1561689925,
        "project": "测试 3",
        "mark": "新增测试流程 3",
    }
    ]
}
```

新增流程返回数据例子:

```
{
  "id": 1001,
  "name": "测试",
  "cuser": 18,
  "user": "fei yao",
  "ctime": 1561688923,
  "utime": 1561689923,
  "project": "测试",
  "mark": "新增测试流程",
}
```

2、获取单个流程，修改流程、删除流程 API

请求方式：GET（查询）PUT（修改）PATCH（修改）DELETE（删除）

请求地址：/api/process/process/1001/

请求地址中 1001 为流程的 id

输入/输出参数：见章节 1 中输入和输出参数说明，修改数据时输入参数均为非必须

返回数据例子:

```
{
  "id": 1001,
  "name": "测试",
  "cuser": 18,
  "user": "san zhang",
  "ctime": 1561688923,
  "utime": 1561689923,
  "project": "测试",
  "mark": "新增测试流程",
},
```

3、保存流程 API

请求方式：POST

请求地址：/api/process/save/

输入参数（新增）：

数据由 body 传入

返回数据例子：

```
{
  "success": true,
}
```

4、提交流程 API

请求方式：POST

请求地址：/api/process/upload/

输入参数（新增）：

参数	数据类型	是否必须	说明
id	string	是	流程 ID

返回数据例子：

```
{
  "success": true,
}
```

5、执行流程 API

请求方式：POST

请求地址：/api/process/execute/

输入参数（新增）：

参数	数据类型	是否必须	说明
id	string	是	流程 ID

返回数据例子：

```
{
  "success": true,
  "job_id": 123,
}
```


6、停止流程 API

请求方式： POST

请求地址： /api/process/stop/123/

请求地址中 123 为 job_id

输入参数（新增）：

参数	数据类型	是否必须	说明
job_id	int	是	job_id

返回数据例子：

```
{
  "success": true,
}
```

7、文件列表 API

请求方式： GET

请求地址： /api/process/file/list/

** 输入参数（查询）： **

参数	数据类型	是否必须	说明
id	string	是	流程 ID

返回数据例子：

```
{
  "name": "aaa",
  "type": "direcotory",
  "children": [
    {
      "name": "bbb",
      "type": "directory",
      "children": [
        {
          "name": "ddd",
          "type": "directory",
          "children": []
        }
      ]
    },
    {
      "name": "ccc",
      "type": "directory",
      "children": [
        {
          "name": "eee.yml",
          "type": "file",

```

(下页继续)

(续上页)

```
        "children": []
      }
    ]
  }
}
```

8、文件保存 API

请求方式： POST

请求地址： /api/process/file/save/

** 输入参数 (新增): **

参数	数据类型	是否必须	说明
id	string	是	流程 ID
path	string	是	需要修改的文件的路径

返回数据例子:

```
{
  "success": true,
}
```

9、文件拷贝 API

请求方式： POST

请求地址： /api/process/file/copy/

** 输入参数 (新增): **

参数	数据类型	是否必须	说明
id	string	是	流程 ID

返回数据例子:

```
{
  "success": true,
}
```

基础模块

说明

- 将基础模块的 yml 格式的文件转成 json 文件

1、基础表列表 API

请求方式：GET（查询）

请求地址：/api/nodes/package/

Content-Type:

新增数据的时候需要指定 Content-Type，以下对 Content-Type 进行说明：

application/x-www-form-urlencoded —— 表示通过表单方式提交
application/json —— 表示传入数据为 json 格式字符串

查询参数：

参数	数据类型	是否必须	说明
name	string	否	函数名
title	string	否	标题
description	string	否	描述

GET 返回数据例子：

```
[
  {
    "title": "程序",
    "description": null,
    "children": [
      {
        "name": "code",
        "className": null,
        "namespace": "fxpa.fxprogramming",
        "title": "python 代码块",
        "nodeType": "function",
        "description": "运行 python 代码块",
        "cls": {
          "icon": null,
          "color": null
        },
        "ports": {
          "input": [
            {
              "name": null,
              "portType": "flow",
              "title": null,
              "dataType": null,
              "value": null
            }
          ]
        }
      }
    ]
  }
]
```

(下页继续)

(续上页)

```

        {
            "name": "content",
            "portType": "input",
            "title": "代码块",
            "dataType": "string",
            "elementType": "textarea",
            "value": null
        }
    ],
    "output": [
        {
            "name": null,
            "portType": "flow",
            "title": null,
            "dataType": null,
            "value": null
        }
    ]
},
"detailPanel":
{
    "common":
    {
        title: 通用设置,
        formDatas: [
            {
                "name": "before",
                "dataType": "number",
                "elementType": "input",
                "value": 0,
                "title": "前置延时 (秒)"
            },
            {
                "name": "after",
                "dataType": "number",
                "elementType": "input",
                "value": 0,
                "title": "后置延时 (秒)"
            },
            {
                "name": "continueOnErr",
                "dataType": "bool",
                "elementType": "checkbox",
                "choices": [
                    True,
                    False
                ],
                "value": False,
                "title": "错误继续执行"
            }
        ]
    }
}
},
]

```

(下页继续)

(续上页)

```
    },  
  ]  
}
```

2、查询 API

请求方式：GET（查询）

请求地址：/api/nodes/package/?search="code"

请求地址中 code 为函数名

输入/输出参数：见章节 1 中输入和输出参数说明，修改数据时输入参数均为非必须

返回数据例子：

```
[  
  {  
    "title": "程序",  
    "description": null,  
    "children": [  
      {  
        "name": "code",  
        "className": null,  
        "namespace": "fxpa.fxprogramming",  
        "title": "python 代码块",  
        "nodeType": "function",  
        "description": "运行 python 代码块",  
        "cls": {  
          "icon": null,  
          "color": null  
        },  
        "ports": {  
          "input": [  
            {  
              "name": null,  
              "portType": "flow",  
              "title": null,  
              "dataType": null,  
              "value": null  
            },  
            {  
              "name": "content",  
              "portType": "input",  
              "title": "代码块",  
              "dataType": "string",  
              "elementType": "textarea",  
              "value": null  
            }  
          ],  
          "output": [  
            {  
              "name": null,  
              "portType": "flow",
```

(下页继续)

变量

说明

- 变量增删改查

1、变量列表、新增变量、批量删除变量 API

请求方式：GET（查询）POST(新增) DELETE（批量删除）

请求地址：/api/process/variable/

Content-Type:

新增数据的时候需要指定 Content-Type，以下对 Content-Type 进行说明：

application/x-www-form-urlencoded —— 表示通过表单方式提交
application/json —— 表示传入数据为 json 格式字符串

查询参数：

参数	数据类型	是否必须	说明
name	string	是	变量名称

输入参数（新增）：

参数	数据类型	是否必须	说明
name	string	是	变量名称
type	string	是	类型
visible	bool	否	是否可见
mark	string	否	备注

输出参数：

参数	数据类型	是否必须	说明
name	string	是	变量名称
type	string	是	类型
visible	bool	否	是否可见
mark	string	否	备注

批量删除参数：

参数	数据类型	是否必须	说明
pk	string	与 pk[] 不能都为空	主键，多个主键用半角逗号隔开。通过 http body 传入
pk[]	array	与 pk 不能都为空	主键数组。通过 http body 传入

排序：

参数	数据类型	是否必须	说明
ordering	string	否	一般默认按 id 倒叙 <ul style="list-style-type: none">ordering=id 表示按 id 排序 ordering=-id 表示按 id 倒叙多个字段排序用半角逗号分隔

GET 返回数据例子:

```
{
  "count": 9,
  "next": null,
  "previous": null,
  "results": [
    {
      "id": 1001,
      "name": "变量 1",
      "type": "password",
      "visible": false,
      "cuser": 18,
      "user": "san zhang",
      "ctime": 1561688923,
      "mark": "新增测试流程",
    },
    {
      "id": 1002,
      "name": "变量 2",
      "type": "password",
      "visible": false,
      "cuser": 18,
      "user": "san zhang",
      "ctime": 1561688923,
      "mark": "新增测试流程",
    },
    {
      "id": 1003,
      "name": "变量 3",
      "type": "password",
      "visible": false,
      "cuser": 18,
      "user": "san zhang",
      "ctime": 1561688923,
      "mark": "新增测试流程",
    }
  ]
}
```

新增流程返回数据例子:

```
{
  "id": 1001,
  "name": "变量 1",
}
```

(下页继续)

(续上页)

```

    "type": "password",
    "visible": false,
    "cuser": 18,
    "user": "san zhang",
    "ctime": 1561688923,
    "mark": "新增测试流程",
  }

```

2、获取单个变量，修改变量、删除变量 API

请求方式：GET（查询）PUT（修改）PATCH（修改）DELETE（删除）

请求地址：/api/process/variable/1001/

请求地址中 1001 为变量的 id

输入/输出参数：见章节 1 中输入和输出参数说明，修改数据时输入参数均为非必须

返回数据例子：

```

{
  "id": 1001,
  "name": "变量 1",
  "type": "password",
  "visible": false,
  "cuser": 18,
  "user": "san zhang",
  "ctime": 1561688923,
  "mark": "新增测试流程",
}

```

流程展示

说明

- 获取已编辑流程的完整数据信息

请求方式：GET

请求地址：/api/process/intact

Content-Type:

application/x-www-form-urlencoded —— 表示通过表单方式提交

查询参数：

参数	数据类型	是否必须	说明
id	string	否	pk(无需输入)

输入参数：

参数	数据类型	是否必须	说明
----	------	------	----

输出参数：

参数	数据类型	是否必须	说明
id	string	是	流程 id
name	string	是	流程名称
nodeList	array	是	节点列表
--id	string	是	节点 id
--class	string	是	类名
--name	string	是	节点函数名称
--namespace	string	是	包名
--title	string	是	节点标题
--nodeType	string	是	节点类型 (功能、事件、变量、流程控制、逻辑运算)
--description	string	是	函数详细描述
--cls	object	是	前端样式
-- --color	string	是	颜色
-- --css	string	是	样式代码
--ports	object	是	端点
-- --in	array	是	入参列表
-- -- --id	string	是	端点 id
-- -- --name	string	是	端点名称
-- -- --title	string	是	端点标题
-- -- --portType	string	是	端点类型 (flow、input)
-- -- --dataType	string	是	数据类型
-- -- --value	string	是	数据值
-- --out	string	是	出参列表

日志查询请求示例：

- Content-Type: 'application/x-www-form-urlencoded'

```
curl http://ip:port/api/process/intact?id=1001
```

日志查询返回数据示例：

- Content-Type: 'application/json'

```
{
  "id": "xxxxxx",
  "name": " 测试",
  "nodeList": [
    {
      "id": "nnnnnn1",
      "className": null,
      "name": "findCoordinate",
      "namespace": "fxpa.fximage",
      "title": " 查找图片坐标",
      "nodeType": "function",
      "x": 400,
      "y": 500,
      "description": " 查找图片在屏幕中的位置，并返回图片中心点在屏幕上的坐标",
      "cls": {
        "icon": null,
        "color": null
      },
      "ports": {
```

(下页继续)

(续上页)

```

        "input": [
            {
                "id": "pppp1",
                "name": null,
                "portType": "flow",
                "title": null,
                "dataType": null,
                "value": null,
                "cls": {
                    "color": null,
                    "css": null
                }
            }
        ],
        "output": [
            {
                "id": "ppppp3",
                "name": null,
                "portType": "flow",
                "title": null,
                "dataType": null,
                "value": null,
                "cls": {
                    "color": null,
                    "css": null
                }
            }
        ]
    },
    "detailPanel": {
        "common": {
            {
                title: 通用设置,
                formDatas: [
                    {
                        "name": "before",
                        "dataType": "number",
                        "elementType": "input",
                        "value": null,
                        "title": "Delay Before (secs) "
                    }
                ]
            }
        },
        "options": {
            title: 选项,
            formDatas: [
                {
                    "name": "confidence",
                    "dataType": "float",
                    "elementType": "input",
                    "value": 0.9,
                    "title": " 置信度"
                }
            ]
        }
    }
}

```

(下页继续)

(续上页)

```
    }
  },
  "linkList": [
    {
      "id": "link-xxxx",
      "source": "nnnnnn1",
      "sourcePort": "ppppp3",
      "target": "nnnnn2",
      "targetPort": "ppppp7",
      "cls": {
        "linkType": null,
        "linkColor": null,
        "linkThickness": null
      }
    }
  ],
  "variables": [
    {
      "name": "imagePath",
      "type": "string",
      "scope": "flow",
      "value": "",
      "cuser": ""
    }
  ]
}
```

日志查询错误返回示例:

```
{
  "detail": "流程获取失败"
}
```

nodeType 分类

事件: event
功能: function
变量设置: variable_set
变量获取: variable_get
流程控制: programming
逻辑运算: operation

日志

说明

- 获取流程调试日志

1、轮询获取，每次指定 row(已读行号) 实现滚动输出

请求方式：GET

请求地址：/api/process/logfile

Content-Type:

application/x-www-form-urlencoded —— 表示通过表单方式提交

查询参数：

参数	数据类型	是否必须	说明
jobId	string	是	作业 id
row	int	是	行号

输入参数：

参数	数据类型	是否必须	说明
	string	否	

输出参数：

参数	数据类型	是否必须	说明
logcontent	string	是	日志内容

日志查询请求示例：

```
/api/process/logfile?jobId=job001&row=0
```

日志查询返回数据示例：

- Content-Type: 'application/json'

```
{
  "logcontent": "2020-11-26 13:29:35.323 NIO | PID | THREAD tsids2 >>>>>>>process_
↪start \n
                2020-11-26 13:29:36.323 NIO | PID | THREAD tsids2 >>>>>>>process_
↪execute...
                2020-11-26 13:29:37.323 NIO | PID | THREAD tsids2 >>>>>>>process_
↪execute...
                2020-11-26 13:29:38.323 NIO | PID | THREAD tsids2 >>>>>>>process_
↪execute...
                2020-11-26 13:29:40.323 NIO | PID | THREAD tsids2 >>>>>>>process_
↪end"
}
```

日志查询错误返回示例：

```
{
  "detail": "流程试运行发生异常"
}
```

2、下载日志文件

请求方式：GET

请求地址：/api/process/logfile/download

Content-Type:

application/x-www-form-urlencoded —— 表示通过表单方式提交

查询参数：

参数	数据类型	是否必须	说明
jobId	string	是	作业 id

输入参数：

参数	数据类型	是否必须	说明
	string	否	

输出参数：

参数	数据类型	是否必须	说明
logcontent	string	否	日志内容

日志下载请求示例：

```
wget http://ip:port/api/process/logfile?jobId=job001
```

日志下载返回数据示例：

- Content-Type: 'application/octet-stream'

返回二进制文件

日志查询错误返回示例：

```
{
  "detail": "日志未找到"
}
```

1.1.5 登录

说明

- 使用 studio 需要先登录

请求方式： POST

请求地址： /api/login/

Content-Type:

application/json —— 表示传入数据为 json 格式字符串

输入参数 (新增):

参数	数据类型	是否必须	说明
username	string	是	用户名
password	string	是	用户密码

输出参数:

参数	数据类型	是否必须	说明
token	string	是	用户凭证

登录请求示例:

```
{
  "username": "admin",
  "password": "admin"
}
```

POST 登录返回数据示例:

```
{
  "token": "faisjdjasidjijas23d98c3fhhc8d8"
}
```

POST 登录错误返回示例:

```
{
  "detail": "The username or password is not correct"
}
```

1.1.6 登出

说明

- 用户主动登出系统，并失效 token

请求方式： POST

请求地址： /api/logout/

Content-Type:

application/json —— 表示传入数据为 json 格式字符串

输入参数:

参数	数据类型	是否必须	说明
token	string	是	用户名

输出参数:

参数	数据类型	是否必须	说明
	string	是	

登出请求示例:

```
{
  "token": "WmZPC08FASHCuo9Ic9edP66bFLwb3ChqVJbBUN1iYEEa8PyCeyWrM3M6SenwzVYM"
}
```

POST 登出返回数据示例:

```
{
}
```

POST 登出错误返回示例:

```
{
  "detail": "The username or password is not correct"
}
```