

新版云推送

作者:chjyao1

更新时间:2019-03-14

1. 默认文档

1.1. 界面交互

账户管理

获取激活步骤

基本信息

接口名称:获取激活步骤

接口地址:/edmccloud/web/php/user/account.php?method=getActvieState

请求方法:GET

请求数据类型:X-WWW-FORM-URLENCODED

响应类型:JSON

状态:有效

接口描述:

激活账户

请求参数

| 参数名称 | 是否必须 | 数据类型 | 默认值 | 描述 |
|-------------|------|--------|-------------------|--------|
| method | true | string | getActvieState | 固定写死 |
| reset_state | true | string | true 重置 false 不重置 | 重置激活状态 |

响应参数

| 参数名称 | 是否必须 | 数据类型 | 描述 |
|--------------------|------|---------|--------------------------------|
| success | true | boolean | 是否成功 |
| data | true | object | |
| active_step | true | number | 激活步骤 1 选择模式 2 激活中 3 激活完成 |
| active_mode | true | string | 激活模式 2 使用系统发信域 1 使用个人发信域 |
| reset_state_succes | true | string | 重置激活状态是否成功 true 成功 false 失败 |

检验域

基本信息

接口名称:检验域

接口地址:/edmcloud/web/php/user/account.php?method=checkDomain 请求

方法:POST

请求数据类型:X-WWW-FORM-URLENCODED

响应类型:JSON

状态:有效

接口描述:

检查域有效性

请求参数

| 参数名称 | 是否必须 | 数据类型 | 默认值 | 描述 |
|------------|------|--------|-----|---------|
| domainName | true | string | | 需要检查的域名 |

响应参数

| 参数名称 | 是否必须 | 数据类型 | 描述 |
|---------|------|---------|--------------------------|
| success | true | boolean | 是否成功 true 成功 false 失败 |
| code | true | number | 错误码 |
| data | true | object | 域各项参数检验结果 |
| MX | true | boolean | |
| SPF | true | boolean | |
| DKIM | true | boolean | |
| DMARC | true | boolean | |

示例数据

错误码对照表:

10002 域不存在

10003 其他错误

- 10005 创建域失败
- 10004 检验域失败
- 10006 域已经是检验通过的

增加域

基本信息

接口名称:增加域

接口地址:/edmccloud/web/php/user/account.php?method=addDomain 请求

方法:POST

请求数据类型:X-WWW-FORM-URLENCODED

响应类型:JSON

状态:有效

接口描述:

增加域

请求参数

| 参数名称 | 是否必须 | 数据类型 | 默认值 | 描述 |
|------------|------|--------|-----|---------|
| domainName | true | string | | 需要检查的域名 |

响应参数

| 参数名称 | 是否必须 | 数据类型 | 描述 |
|---------|------|---------|--------------------------|
| success | true | boolean | 是否成功 true 成功 false 失败 |
| code | true | number | 错误码 |

示例数据

- 错误码对照表:
- 10002 域不存在
 - 10003 其他错误
 - 10005 创建域失败
 - 10004 检验域失败
 - 10006 域已经是检验通过的
 - 10007 域总数超过用户配额

列出所有域

基本信息

接口名称:列出所有域

接口地址:/edmccloud/web/php/user/account.php?method=listDomainForAccount 请求

方法:GET

请求数据类型:X-WWW-FORM-URLENCODED

响应类型:JSON

状态:有效

接口描述:

列出所有域

请求参数

| 参数名称 | 是否必须 | 数据类型 | 默认值 | 描述 |
|---------------|------|---------------|-------------|--------|
| success | true | boolean | | |
| data | true | array[object] | | 域数据列表 |
| uid | true | string | | 域唯一 id |
| domain_name | true | string | | 域名 |
| create_time | true | string | | 创建时间 |
| is_activation | true | string | 0 没验证 1 已验证 | 是否验证 |
| accountid | true | string | | 账户 id |
| used | true | boolean | | 使用使用中 |

示例数据

错误码对照表:

- 10002 域不存在
- 10003 其他错误
- 10005 创建域失败
- 10004 检验域失败
- 10006 域已经是检验通过的

修改域

基本信息

接口名称:修改域

接口地址:/edmcloud/web/php/user/account.php?method=updateDomain 请求

方法:POST

请求数据类型:X-WWW-FORM-URLENCODED

响应类型:JSON

状态:有效

接口描述:

修改域名 同事同步修改云端域名

请求参数

| 参数名称 | 是否必须 | 数据类型 | 默认值 | 描述 |
|----------|------|--------|-----|--------|
| domainid | true | string | | 修改域 ID |
| newName | true | string | | 变更后域名 |

响应参数

| 参数名称 | 是否必须 | 数据类型 | 描述 |
|---------|------|---------|--------------------------|
| success | true | boolean | 是否成功 true 成功 false 失败 |
| code | true | number | 错误码 |

示例数据

错误码对照表:

- 10011 禁止修改已经验证域
- 10010 和其他域重名
- 10002 域不存在
- 10003 其他错误
- 10005 创建域失败
- 10004 检验域失败
- 10006 域已经是检验通过的
- 10007 域总数超过用户配额

触发邮件

激活账户

基本信息

接口名称:激活账户

接口地址:/edmccloud/web/php/user/account.php

请求方法:GET

请求数据类型:X-WWW-FORM-URLENCODED

响应类型:JSON

状态:有效

接口描述:
激活账户

请求参数

| 参数名称 | 是否必须 | 数据类型 | 默认值 | 描述 |
|------------|------|--------|--------|------|
| domainName | true | string | | 域名 |
| method | true | string | active | 固定写死 |

响应参数

| 参数名称 | 是否必须 | 数据类型 | 描述 |
|---------|------|--------|--------------------------|
| success | true | string | 是否成功 true 成功 false 失败 |
| reason | true | string | 失败原因 |

使用系统提供域进行激活

基本信息

接口名称:使用系统提供域进行激活

接口地址:/edmccloud/web/php/user/account.php?method=activeUseSystemDomain

请求方法:GET

请求数据类型:X-WWW-FORM-URLENCODED

响应类型:JSON

状态:有效

接口描述:

激活账户

响应参数

| 参数名称 | 是否必须 | 数据类型 | 描述 |
|---------|------|---------|--------------|
| success | true | boolean | 是否成功 |
| data | true | object | |
| code | true | string | 错误码 参见错误码对照表 |

示例数据

注意如果此激活成功

请调用获取激活步骤接口

列出所有用户的发信域

基本信息

接口名称:列出所有用户的发信域

接口地址:/edmccloud/web/php/admin/account_adm.php?method=listAllUserDomain

请求方法:GET

请求数据类型:X-WWW-FORM-URLENCODED

响应类型:JSON

状态:有效

接口描述:

列出所有域

请求头

| 参数名称 | 是否必须 | 默认值 | 描述 |
|----------|-------|-----|-------------|
| account | false | | 域的所有者账户名 |
| domain | false | | 域名检索条件 模糊匹配 |
| pageno | true | 1 | 页号 |
| pagesize | false | 10 | 每页数据条数 |

请求参数

| 参数名称 | 是否必须 | 数据类型 | 默认值 | 描述 |
|---------------|------|---------------|-------------|--------|
| success | true | boolean | | |
| data | true | array[object] | | 域数据列表 |
| uid | true | string | | 域唯一 id |
| domain_name | true | string | | 域名 |
| create_time | true | string | | 创建时间 |
| is_activation | true | string | 0 没验证 1 已验证 | 是否验证 |
| accountid | true | string | | 账户 id |
| used | true | boolean | | 使用使用中 |

响应参数

| 参数名称 | 是否必须 | 数据类型 | 描述 |
|---------------|------|---------------|-------------------------|
| success | true | boolean | 是否调用成功 true 成功 false 失败 |
| reason | true | string | 失败原因 不用理会 |
| pagestart | true | number | 本业第一条行号 |
| pageend | true | number | 本页最后一条行号 |
| totalrow | true | string | 总行数 |
| totalpage | true | number | 总页数 |
| pagedata | true | array[object] | 本页数据列表 |
| uid | true | string | 域唯一 ID |
| domain_name | true | string | 域名 |
| create_time | true | string | 创建时间 |
| is_activation | true | string | 是否检验通过 |
| accountid | true | string | 账户 ID |
| domain_type | true | string | 域类型 |
| used_count | true | string | 域使用总数 |
| account | true | string | 账户名 |

示例数据

```
{
  "success": true,
  "reason": "",
  "pagestart": 1,
  "pageend": 5,
  "totalrow": "6",
  "totalpage": 2,
  "pagedata": [
    {
      "uid": "sp_domain_5c6380449d6a9",
      "domain_name": "mail-cloud.cn",
      "create_time": "2019-02-13 10:26:12",
      "is_activation": "1",
      "accountid": "1200",
      "domain_type": "1",
      "used_count": "0",
      "account": "dandan1@xin.cn"
    }
  ]
}
```

```
},
{
  "uid": "sp_domain_5c638955350aa",
  "domain_name": "mytest.com",
  "create_time": "2019-02-13 11:04:53",
  "is_activation": "0",
  "accountid": "1202",
  "domain_type": "1",
  "used_count": "0",
  "account": "ycj1@xin.cn"
},
{
  "uid": "sp_domain_5c638966ef29b",
  "domain_name": "mail-cloud.cn",
  "create_time": "2019-02-13 11:05:10",
  "is_activation": "1",
  "accountid": "1202",
  "domain_type": "1",
  "used_count": "1",
  "account": "ycj1@xin.cn"
},
{
  "uid": "sp_domain_5c665fe91f9b3",
  "domain_name": "mail-cloud.cn",
  "create_time": "2019-02-15 14:44:57",
  "is_activation": "1",
  "accountid": "1214",
  "domain_type": "1",
  "used_count": "0",
  "account": "ycjtest1@xin.cn"
}
]
```

列出用户激活信息

基本信息

接口名称:列出用户激活信息

接口地址:/edmccloud/web/php/admin/account_adm.php?method=getAccountActiveInfo

请求方法:GET

请求数据类型:X-WWW-FORM-URLENCODED

响应类型:JSON

状态:有效

接口描述:

请求参数

| 参数名称 | 是否必须 | 数据类型 | 默认值 | 描述 |
|---------|------|--------|-----|--------|
| account | true | string | | 用户登录账号 |

响应参数

| 参数名称 | 是否必须 | 数据类型 | 描述 |
|---------------------|------|---------|-------------|
| success | true | boolean | 是否调用成功 |
| actived | true | boolean | 是否激活 |
| data | true | object | 激活信息 |
| uid | true | string | |
| tr_api_user | true | string | 触发邮件 API 标识 |
| tr_api_key | true | string | 触发邮件 API 秘钥 |
| create_date | true | string | |
| platform | true | string | |
| domain_id | true | string | 域 ID |
| accountid | true | string | |
| is_activation | true | string | 域是否检验通过 |
| type | true | string | |
| account | true | string | |
| parent_sp_account | true | string | |
| bt_api_user | true | string | 批量邮件 API 标识 |
| bt_api_key | true | string | 批量邮件 API 秘钥 |
| product_type | true | string | |
| is_default | true | string | |
| domain_name | true | string | 域名 |
| domain_type | true | string | |
| domain_type_display | true | string | 域类型显示文字 |

示例数据

```
{
  "success": true,
  "actived": true,
  "data": {
    "uid": "sp_account_5c6ba8f4630c9",
    "tr_api_user": "762821440.1275.tr",
    "tr_api_key": "3eA2ic3RF5byNwXO",
    "create_date": "2019-02-19 14:57:56",
    "platform": "sendcloud",
    "domain_id": "sp_sys_domain_5c6ba8e831ce7",
    "accountid": "1275",
    "is_activation": "1",
    "type": "1",
    "account": "dandan1@xin.cn",
    "parent_sp_account": null,
    "bt_api_user": "762821440.1275.bt",
    "bt_api_key": "8tcxzz0aPYOQfUpK",
    "product_type": "p_smart",
```

```
"is_default": "1",

"domain_name": "mail-cloud.cn",
"domain_type": "2",
"domain_type_display": "\u516c\u6709"
}
}
```

获取域配置信息

基本信息

接口名称:获取域配置信息

接口地址:/edmccloud/web/php/user/account.php?method=getDomainInfo 请求

方法:POST

请求数据类型:X-WWW-FORM-URLENCODED

响应类型:JSON

状态:有效

接口描述:

检查域有效性

请求参数

| 参数名称 | 是否必须 | 数据类型 | 默认值 | 描述 |
|------------|------|--------|-----|---------|
| domainName | true | string | | 需要获取的域名 |

响应参数

| 参数名称 | 是否必须 | 数据类型 | 描述 |
|--------------|------|---------|------------|
| success | true | boolean | 是否获取信息成功 |
| data | true | object | 域名配置信息 |
| name | true | string | 域名 |
| type | true | string | 域类型 |
| verify | true | number | 域名配置项合格度评分 |
| spf.domain | true | string | spf 发信域 |
| spf.value | true | string | spf 期望正确项 |
| dkim.domain | true | string | dkim 域名 |
| dkim.value | true | string | dkim 正确配置值 |
| mx.domain | true | string | mx 域名 |
| mx.value | true | string | mx 正确配置值 |
| dmarc.domain | true | string | drmc 域名 |
| dmarc.value | true | string | drmc 正确配置值 |
| gmtCreated | true | string | 创建时间 |
| gmtUpdated | true | string | 更新时间 |

示例数据

错误码对照表:

- 10002 域不存在
- 10003 其他错误
- 10005 创建域失败
- 10004 检验域失败
- 10006 域已经是检验通过的

获取域配置信息管理员端

基本信息

接口名称:获取域配置信息管理员端

接口地址:/edmccloud/web/php/user/account.php?method=getDomainInfo 请求

方法:POST

请求数据类型:X-WWW-FORM-URLENCODED

响应类型:JSON

状态:有效

接口描述:

检查域有效性

请求参数

| 参数名称 | 是否必须 | 数据类型 | 默认值 | 描述 |
|------------|------|--------|-----|---------|
| domainName | true | string | | 需要获取的域名 |

响应参数

| 参数名称 | 是否必须 | 数据类型 | 描述 |
|--------------|------|---------|------------|
| success | true | boolean | 是否获取信息成功 |
| data | true | object | 域名配置信息 |
| name | true | string | 域名 |
| type | true | string | 域类型 |
| verify | true | number | 域名配置项合格度评分 |
| spf.domain | true | string | spf 发信域 |
| spf.value | true | string | spf 期望正确项 |
| dkim.domain | true | string | dkim 域名 |
| dkim.value | true | string | dkim 正确配置值 |
| mx.domain | true | string | mx 域名 |
| mx.value | true | string | mx 正确配置值 |
| dmarc.domain | true | string | drmc 域名 |
| dmarc.value | true | string | drmc 正确配置值 |
| gmtCreated | true | string | 创建时间 |
| gmtUpdated | true | string | 更新时间 |

示例数据

错误码对照表:

- 10002 域不存在
- 10003 其他错误
- 10005 创建域失败
- 10004 检验域失败
- 10006 域已经是检验通过的

删除域

基本信息

接口名称:删除域

接口地址:/edmccloud/web/php/user/account.php?method=removeDomain 请求

方法:POST

请求数据类型:X-WWW-FORM-URLENCODED

响应类型:JSON

状态:有效

接口描述:

增加域

请求参数

| 参数名称 | 是否必须 | 数据类型 | 默认值 | 描述 |
|----------|------|--------|-----|--------|
| domainid | true | string | | 修改域 ID |

响应参数

| 参数名称 | 是否必须 | 数据类型 | 描述 |
|---------|------|---------|-----------------------|
| success | true | boolean | 是否成功 true 成功 false 失败 |
| code | true | string | 在删除域失败时候错误码 |

示例数据

可能返回错误码:

- 10013 被删除域已经激活
- 10008 删除失败

API 管理

加载 API 安全配置

基本信息

接口名称:加载 API 安全配置

接口地址:/edmcloud/web/php/user/api_config.php?method=loadApiSecurity

请求方法:GET

请求数据类型:X-WWW-FORM-URLENCODED

响应类型:JSON

状态:有效

接口描述:

获取 API 认证信息

响应参数

| 参数名称 | 是否必须 | 数据类型 | 描述 |
|------------|------|--------|--------------|
| code | true | number | 状态码 参见状态码对照表 |
| data | true | object | 接口认证配置信息 |
| uid | true | string | |
| account | true | string | 账户名 |
| api_key | true | string | 用户访问 API 令牌 |
| api_secret | true | string | 密钥 |
| algorithm | true | string | 算法 |
| account_id | true | string | 账户 ID |

状态码对照表

404 服务没找到

500 操作错误

403 无访问权限

200 请求成功

416 请求内容有错误

保存安全配置

基本信息

接口名称:保存安全配置

接口地址:/edmcloud/web/php/user/api_config.php?method=saveApisecurity

请求方法:GET

请求数据类型:X-WWW-FORM-URLENCODED

响应类型:JSON

状态:有效

接口描述:

获取 API 认证信息

请求参数

| 参数名称 | 是否必须 | 数据类型 | 默认值 | 描述 |
|---------|-------|--------|--------|------|
| api_key | false | string | 用户账户名称 | 访客标识 |

| | | | | |
|------------|------|--------|--|----------|
| api_secret | true | string | | Api 秘钥明文 |
|------------|------|--------|--|----------|

响应参数

| 参数名称 | 是否必须 | 数据类型 | 描述 |
|------|------|--------|----|
| code | true | number | |

示例数据

错误相应

```
{
  "code": 500,
  "reason": "操作错误"
}
```

生成秘钥

基本信息

接口名称:生成秘钥

接口地址:/edmccloud/web/php/user/api_config.php?method=generalSecretKey

请求方法:GET

请求数据类型:X-WWW-FORM-URLENCODED

响应类型:JSON

状态:有效

接口描述:

获取 API 认证信息

响应参数

| 参数名称 | 是否必须 | 数据类型 | 描述 |
|------------|------|--------|--------|
| code | true | number | 状态码 |
| data | true | object | |
| secret_key | true | string | 新生成的秘钥 |

触发邮件任务管理

列出所有发送任务

基本信息

接口名称:列出所有发送任务

接口地址:/edmccloud/web/php/user/triggermailtask.php?method=list_task 请求

方法:POST

请求数据类型:X-WWW-FORM-URLENCODED

响应类型:JSON

状态:有效

接口描述:

分页列出所有触发类有邮件发送任务

请求参数

| 参数名称 | 是否必须 | 数据类型 | 默认值 | 描述 |
|-------------|-------|--------|-----------|----|
| page | true | string | 页号 | |
| size | true | string | 每页数据条数 | |
| searchParam | false | string | 任务主题包含的文字 | |

响应参数

| 参数名称 | 是否必须 | 数据类型 | 描述 |
|----------------|------|---------------|------------------------------|
| status | true | string | 请求是否成功 success 成功 fail 报错 |
| data | true | object | 每页任务列表数据 |
| totalpage | true | number | 总页数 |
| totalrow | true | string | 总数据条数 |
| pageno | true | string | 页码 |
| data | true | array[object] | 本页所有任务列表数据 |
| id | true | string | 任务 ID |
| delive_status | true | string | 投递状态 1 投递成功 2 投递失败 3 无效 4 未知 |
| taskname | true | string | 任务名称 |
| task_describe | true | string | 任务描述 |
| account_id | true | string | 账户 ID |
| account | true | string | 账户名称 |
| send_time | true | string | 发送时间 |
| end_time | true | string | 完成事件 |
| toemail | true | string | 接收邮件地址 |
| tpl_id | true | string | 模板 ID |
| status | true | string | 任务状态 1 未完成 2 完成 |
| from_email | true | string | 发件人邮件地址 |
| domainid | true | string | 发信域 ID |
| click_count | true | string | 点击总数 |
| open_count | true | string | 阅读总数 |
| group_id | true | string | 组 ID |
| group_name | true | string | 组名称 |
| subject | true | string | 主题 |
| complete_count | true | string | 完成地址总数 |
| send_count | true | string | 发送邮件总数 |
| replyto | true | string | 回复地址 |
| fail_count | true | string | 失败地址总数 |
| invalid_count | true | string | 无效地址总数 |
| success_count | true | string | 成功地址总数 |
| tpl_name | true | string | 模板名称 |

示例数据

```
{
  "status": "success",
  "data": {
    "totalpage": 1,
    "totalrow": "8",
    "pageno": "1",
    "data": [
      {
        "id": "1087911007126425600",
        "taskname": "\u6d4b\u8bd5\u53d1\u9001\u8d3a\u5361!",
        "task_describe":
"\u7528\u6237\u901a\u8fc7\u63a5\u53e3\u53d1\u9001\u63a5\u53e3\u53d1\u9001\u89e6\u53d1\u90ae\u4ef6(\u6d4b\u8bd5\u89e6\u53d1\u90ae\u4ef6)",
        "account_id": "1182",
        "account": "ycj12@xin.cn",
        "send_time": "2019-01-23 11:11:29",
        "end_time": "2019-01-23 11:11:34",
        "toemail": null,
        "tpl_id": "2891",
        "status": "2",
        "from_email": "testsend@aa.com",
        "domainid": null,
        "click_count": "0",
        "open_count": "0",
        "group_id": null,
        "group_name": null,
        "subject": "\u6d4b\u8bd5\u89e6\u53d1\u90ae\u4ef6",
        "complete_count": "1",
        "send_count": "1",
        "replyto": null,
        "fail_count": "0",
        "invalid_count": "0",
        "success_count": "1",
        "tpl_name": "\u6d4b\u8bd5\u89e6\u53d1\u90ae\u4ef6",
        "delive_status": "1"
      },
      {
        "id": "1087911464393641984",
        "taskname": "\u6d4b\u8bd5\u53d1\u9001\u8d3a\u5361!",
        "task_describe":
"\u7528\u6237\u901a\u8fc7\u63a5\u53e3\u53d1\u9001\u63a5\u53e3\u53d1\u9001\u89e6\u53d1\u90ae\u4ef6(\u6d4b\u8bd5\u89e6\u53d1\u90ae\u4ef6)",
        "account_id": "1182",
        "account": "ycj12@xin.cn",
        "send_time": "2019-01-23 11:13:18",
        "end_time": null,
      }
    ]
  }
}
```



```
"toemail": null,

"tpl_id": "2891",
"status": "1",
"from_email": "testsend@aa.com",
"domainid": null,
"click_count": "0",
"open_count": "0",
"group_id": null,
"group_name": null,
"subject": "\u6d4b\u8bd5\u89e6\u53d1\u90ae\u4ef6",
"complete_count": "0",
"send_count": "5",
"replyto": null,
"fail_count": "0",
"invalid_count": "0",
"success_count": "0",
"tpl_name": "\u6d4b\u8bd5\u89e6\u53d1\u90ae\u4ef6",
"delive_status": 4
},
{
  "id": "1087911747651768320",
  "taskname": "\u6d4b\u8bd5\u53d1\u9001\u8d3a\u5361!",
  "task_describe":
"\u7528\u6237\u901a\u8fc7\u63a5\u53e3\u53d1\u9001\u63a5\u53e3\u53d1\u9001\u89e6\u53d1\u90ae\u4ef6(\u6d4b\u8bd5\u89e6\u53d1\u90ae\u4ef6)",
  "account_id": "1182",
  "account": "ycj12@xin.cn",
  "send_time": "2019-01-23 11:14:25",
  "end_time": null,
  "toemail": null,
  "tpl_id": "2891",
  "status": "1",
  "from_email": "testsend@aa.com",
  "domainid": null,
  "click_count": "0",
  "open_count": "0",
  "group_id": null,
  "group_name": null,
  "subject": "\u6d4b\u8bd5\u89e6\u53d1\u90ae\u4ef6",
  "complete_count": "0",
  "send_count": "5",
  "replyto": null,
  "fail_count": "0",
  "invalid_count": "0",
  "success_count": "0",
  "tpl_name": "\u6d4b\u8bd5\u89e6\u53d1\u90ae\u4ef6",
  "delive_status": 4
},
```

```
{
  "id": "1087912046525288448",
  "taskname": "\u6d4b\u8bd5\u53d1\u9001\u8d3a\u5361!",
  "task_describe":
"\u7528\u6237\u901a\u8fc7\u63a5\u53e3\u53d1\u9001\u63a5\u53e3\u53d1\u9001\u89e6\u53d1\u90ae\u4ef6(\u6d4b\u8bd5\u89e6\u53d1\u90ae\u4ef6)",
  "account_id": "1182",
  "account": "ycj12@xin.cn",
  "send_time": "2019-01-23 11:15:31",
  "end_time": null,
  "toemail": null,
  "tpl_id": "2891",
  "status": "1",
  "from_email": "testsend@aa.com",
  "domainid": null,
  "click_count": "0",
  "open_count": "0",
  "group_id": null,
  "group_name": null,
  "subject": "\u6d4b\u8bd5\u89e6\u53d1\u90ae\u4ef6",
  "complete_count": "0",
  "send_count": "5",
  "replyto": null,
  "fail_count": "0",
  "invalid_count": "0",
  "success_count": "0",
  "tpl_name": "\u6d4b\u8bd5\u89e6\u53d1\u90ae\u4ef6",
  "delive_status": 4
},
{
  "id": "1087912258329251840",
  "taskname": "\u6d4b\u8bd5\u53d1\u9001\u8d3a\u5361!",
  "task_describe":
"\u7528\u6237\u901a\u8fc7\u63a5\u53e3\u53d1\u9001\u63a5\u53e3\u53d1\u9001\u89e6\u53d1\u90ae\u4ef6(\u6d4b\u8bd5\u89e6\u53d1\u90ae\u4ef6)",
  "account_id": "1182",
  "account": "ycj12@xin.cn",
  "send_time": "2019-01-23 11:16:27",
  "end_time": null,
  "toemail": null,
  "tpl_id": "2891",
  "status": "1",
  "from_email": "testsend@aa.com",
  "domainid": null,
  "click_count": "0",
  "open_count": "0",
  "group_id": null,
  "group_name": null,
```

```
"subject": "\u6d4b\u8bd5\u89e6\u53d1\u90ae\u4ef6",
"complete_count": "0",
"send_count": "5",
"replyto": null,
"fail_count": "0",
"invalid_count": "0",
"success_count": "0",
"tpl_name": "\u6d4b\u8bd5\u89e6\u53d1\u90ae\u4ef6",
"delive_status": 4
},
{
  "id": "1087912502664237056",
  "taskname": "\u6d4b\u8bd5\u53d1\u9001\u8d3a\u5361!",
  "task_describe":
"\u7528\u6237\u901a\u8fc7\u63a5\u53e3\u53d1\u9001\u63a5\u53e3\u53d1\u9001\u89e6\u53d1\u90ae\u4ef6(\u6d4b\u8bd5\u89e6\u53d1\u90ae\u4ef6)",
  "account_id": "1182",
  "account": "ycj12@xin.cn",
  "send_time": "2019-01-23 11:17:25",
  "end_time": null,
  "toemail": null,
  "tpl_id": "2891",
  "status": "1",
  "from_email": "testsend@aa.com",
  "domainid": null,
  "click_count": "0",
  "open_count": "0",
  "group_id": null,
  "group_name": null,
  "subject": "\u6d4b\u8bd5\u89e6\u53d1\u90ae\u4ef6",
  "complete_count": "0",
  "send_count": "5",
  "replyto": null,
  "fail_count": "0",
  "invalid_count": "0",
  "success_count": "0",
  "tpl_name": "\u6d4b\u8bd5\u89e6\u53d1\u90ae\u4ef6",
  "delive_status": 4
},
{
  "id": "1087912771531706368",
  "taskname": "\u6d4b\u8bd5\u53d1\u9001\u8d3a\u5361!",
  "task_describe":
"\u7528\u6237\u901a\u8fc7\u63a5\u53e3\u53d1\u9001\u63a5\u53e3\u53d1\u9001\u89e6\u53d1\u90ae\u4ef6(\u6d4b\u8bd5\u89e6\u53d1\u90ae\u4ef6)",
  "account_id": "1182",
  "account": "ycj12@xin.cn",
  "send_time": "2019-01-23 11:18:29",
```

```
"end_time": "2019-01-23 11:18:48",

"toemail": null,
"tpl_id": "2891",
"status": "2",
"from_email": "testsend@aa.com",
"domainid": null,
"click_count": "0",
"open_count": "0",
"group_id": null,
"group_name": null,
"subject": "\u6d4b\u8bd5\u89e6\u53d1\u90ae\u4ef6",
"complete_count": "5",
"send_count": "5",
"replyto": null,
"fail_count": "0",
"invalid_count": "3",
"success_count": "2",
"tpl_name": "\u6d4b\u8bd5\u89e6\u53d1\u90ae\u4ef6",
"delive_status": 4
},
{
  "id": "1087913637575790592",
  "taskname": "\u6d4b\u8bd5\u53d1\u9001\u8d3a\u5361!",
  "task_describe":
"\u7528\u6237\u901a\u8fc7\u63a5\u53e3\u53d1\u9001\u63a5\u53e3\u53d1\u9001\u89e6\u53d1\u90ae\u4ef6(\u6d4b\u8bd5\u89e6\u53d1\u90ae\u4ef6)",
  "account_id": "1182",
  "account": "ycj12@xin.cn",
  "send_time": "2019-01-23 11:21:56",
  "end_time": "2019-01-23 11:22:25",
  "toemail": null,
  "tpl_id": "2891",
  "status": "2",
  "from_email": "testsend@aa.com",
  "domainid": null,
  "click_count": "0",
  "open_count": "0",
  "group_id": null,
  "group_name": null,
  "subject": "\u6d4b\u8bd5\u89e6\u53d1\u90ae\u4ef6",
  "complete_count": "5",
  "send_count": "5",
  "replyto": null,
  "fail_count": "0",
  "invalid_count": "3",
  "success_count": "2",
  "tpl_name": "\u6d4b\u8bd5\u89e6\u53d1\u90ae\u4ef6",
  "delive_status": 4
```

```
}  
]  
}  
}
```

1.2. 错误码对照表

错误码对照表:

10011 禁止修改已经验证域
10010 和其他域重名
10002 域不存在
10003 其他错误
10005 创建域失败
10004 检验域失败
10006 域已经是检验通过的
10007 域总数超过用户配额
10012 没有可用的系统发信域

1.3. 常量

1.4. 合作伙伴接入

接口认证规则

所有第三方客户接口认证规则如下:

首先第三方客户如果需要调用我们文档需要从界面获取 **apiKey** 和 **apisecret** 用户认证密钥对,**apiKey** 用于标明调用方的身份, **apisecret** 用于加密算法对数据进行混淆(参见 DES 加密算法规则).如何获取密钥对:

首先从我们用户端网站,进入界面 **API 接入->API 认证**

在网站右侧工作区显示的是生成密钥对操作界面,点击"生成密钥"按钮随机生成一个密钥,点击"保存"保存密钥对.

API 用户标识就是你的 apiKey

API 密钥就是您的 apisecret

将密钥对保存到配置文件中,以便您调用我们接口对数据进行签名,如果您调用我们接口没有携带签名信息,或者携带错误的签名信息都将收到认证失败的报错.

现在我们讲解如何在客户端签名方法:

首先我们接口所有提交数据都是以 **POST** 方式提交,并且数据的格式是 **JSON**。

用于认证的参数必须在调用接口 **URL** 中. 以 **GET** 方式发送给后端服务器.

举例说明:

假如我们调用启动触发任务接口:

URL 如下:

http://localhost/edmcloud/api/api_route.php?method=startTriggerMailTask

&sign=%2BgjSPprO1VzXtX0IUt%2FQHM5ilz6%2Fs45ln5tKr8FOoR9MVbasoh%2Bjmg%3D%3D

&apiKey=ycj1%40xin.cn

提交内容如下:

```
{ "taskname": "节日快乐", "subject": "节日快乐 3", "domainid": null, "recipients": "FFAA@qq.com", "group":  
"", "content_tpl": "2977", "fromname": "testsend@xxxx.com", "reply": "tt@aa.com", "need_receipt": false,  
"tpl_vars": [ { "sex": "先生", "name": "usera" } ] }
```

首先参数 **apiKey** 就是您从界面上获取的 **apiKey** **sign** 参数就是您提交内容签名

下面介绍如何对数据进行签名:

首先对您 **POST** 内容进行 **MD5**

生成类似(a84f6e774c18e81c636b0329a60c010b)这样的字符串

然后使用 **DES** 算法,使用您的 **appsecret** 进行加盐,对 **MD5** 字符串进行加密,此时应该返回一个二进制字节数组,然后对这个二进制数组进行 **base64** 编码.

然后会生成这样一个字符串(+gjSPprO1VzXtX0IUt/QHM5ilz6/s45ln5tKr8FOoR9MVbasoh+jmg==),然后将这个字符串通过 **sign** 参数传到服务器即可.

下面附上一个工具类:

DES 类 (用户加密)

```
package cn.xinnet.edmcloudsdk.utils; import java.io.IOException; import
java.io.UnsupportedEncodingException; import java.security.SecureRandom; import javax.crypto.Cipher;
import javax.crypto.SecretKey; import javax.crypto.SecretKeyFactory; import javax.crypto.spec.DESKeySpec;
import org.apache.commons.codec.binary.Base64; public class DES { private byte[] desKey; public DES(String
desKey) { this.desKey = desKey.getBytes(); } public byte[] desEncrypt(byte[] plainText) throws Exception {
SecureRandom sr = new SecureRandom(); byte rawKeyData[] = desKey; DESKeySpec dks = new
DESKeySpec(rawKeyData); SecretKeyFactory keyFactory = SecretKeyFactory.getInstance("DES"); SecretKey
key = keyFactory.generateSecret(dks); Cipher cipher = Cipher.getInstance("DES");
cipher.init(Cipher.ENCRYPT_MODE, key, sr); byte data[] = plainText; byte encryptedData[] =
cipher.doFinal(data); return encryptedData; } public byte[] desDecrypt(byte[] encryptText) throws Exception {
SecureRandom sr = new SecureRandom(); byte rawKeyData[] = desKey; DESKeySpec dks = new
DESKeySpec(rawKeyData); SecretKeyFactory keyFactory = SecretKeyFactory.getInstance("DES"); SecretKey
key = keyFactory.generateSecret(dks); Cipher cipher = Cipher.getInstance("DES");
cipher.init(Cipher.DECRYPT_MODE, key, sr); byte encryptedData[] = encryptText; byte decryptedData[] =
cipher.doFinal(encryptedData); return decryptedData; } public String encrypt(String input) throws Exception {
return base64Encode(desEncrypt(input.getBytes())); } public String decrypt(String input) throws Exception {
byte[] result = base64Decode(input); return new String(desDecrypt(result)); } public static String
base64Encode(byte[] s) throws UnsupportedEncodingException { if (s == null) return null; return new
String(Base64.encodeBase64(s), UTF_8); } private static final String UTF_8 = "UTF-8"; public static byte[]
base64Decode(String s) throws IOException { if (s == null) return null; byte[] b = null; b =
Base64.decodeBase64(s); return b; } public static void main(String[] args) throws Exception { String key =
"abcdefgh"; String input = "a"; DES crypt = new DES(key); System.out.println("Encode:" + crypt.encrypt(input));
System.out.println("Decode:" + crypt.decrypt(crypt.encrypt(input))); } }
```

WsClientHelper 用于提交 **HTTP** 请求

```
package cn.xinnet.edmcloudsdk.client; import java.io.ByteArrayOutputStream; import java.io.File; import
java.net.URLEncoder; import java.util.ArrayList; import java.util.List; import java.util.Map; import java.util.Set;
import org.apache.commons.codec.digest.DigestUtils; import org.apache.commons.lang3.StringUtils; import
org.apache.commons.logging.Log; import org.apache.commons.logging.LogFactory; import
org.apache.http.HttpEntity; import org.apache.http.HttpStatus; import org.apache.http.NameValuePair; import
org.apache.http.client.config.RequestConfig; import org.apache.http.client.entity.UrlEncodedFormEntity; import
org.apache.http.client.methods.CloseableHttpResponse; import org.apache.http.client.methods.HttpPost;
import org.apache.http.entity.StringEntity; import org.apache.http.entity.mime.HttpMultipartMode; import
org.apache.http.entity.mime.MultipartEntity; import org.apache.http.entity.mime.MultipartEntityBuilder; import
org.apache.http.entity.mime.content.FileBody; import org.apache.http.entity.mime.content.StringBody; import
org.apache.http.impl.client.CloseableHttpClient; import org.apache.http.impl.client.HttpClients;
```

```

import org.apache.http.message.BasicNameValuePair; import org.apache.http.util.EntityUtils; import
cn.xinnet.edmcloudsdk.domain.CallResult; import cn.xinnet.edmcloudsdk.utils.DES; import
com.google.gson.Gson; import com.google.gson.GsonBuilder; public class WsClientHelper { // 是否处于开发模
式(上线时注意将值改为 false) private static boolean DEV_MODE = true; private static Log log =
LogFactory.getLog(WsClientHelper.class); public static String generalContentSign(String content, String
appsecret) { String credential = null; try { String contentmd5 = DigestUtils.md5Hex(content);
log.info("=====>内容 MD5:" + contentmd5); DES destool = new DES(appsecret); credential =
destool.encrypt(contentmd5); log.info(StringUtils.repeat("=", 100)); log.info("\n 加密秘钥:" + appsecret); log.info("
加密原文:" + content.toString()); log.info("生成签名:" + credential); } catch (Exception e) { log.error("生成接口签
名失败", e); } return credential; } private static String encodeJson(Object obj) { Gson gs = null; gs = new
GsonBuilder().serializeNulls().setPrettyPrinting().setDateFormat("yyyy-MM-dd HH:mm:sss").create(); String
jsonstr = gs.toJson(obj); return jsonstr; } public static CallResult callWs(String url, String appkey, String
appsecret, Object bizparams, String method) { CallResult cr = new CallResult(); cr.setSuccess(false);
StringEntity postentity = null; String respstr = ""; List<NameValuePair> queryparams = null; HttpPost httpPost =
null; try { CloseableHttpClient httpClient = HttpClients.createDefault(); // 创建请求 URL String query = ""; String
sendContent = ""; sendContent = encodeJson(bizparams); System.out.println("\n send json post:\n" +
sendContent); // 控制参数 String sign = generalContentSign(sendContent, appsecret); queryparams = new
ArrayList<NameValuePair>(); queryparams.add(new BasicNameValuePair("method", method));
queryparams.add(new BasicNameValuePair("sign", sign)); queryparams.add(new
BasicNameValuePair("apiKey", appkey)); UrlEncodedFormEntity queryBuilder = new UrlEncodedFormEntity(
queryparams); ByteArrayOutputStream bout = new ByteArrayOutputStream(); queryBuilder.writeTo(bout);
bout.flush(); query = new String(bout.toByteArray(), "utf-8"); bout.close(); if (DEV_MODE) { url = url + "?" +
query + "&XDEBUG_SESSION_START=ECLIPSE_DBG&KEY=15022624739804"; } else { url = url + "?" +
query; } log.info("\n\n\n 处理后的请求 url:" + url); httpPost = new HttpPost(url); // 设置请求性能参数
RequestConfig requestConfig = RequestConfig.custom()
.setSocketTimeout(115000).setConnectTimeout(115000)
.setMaxRedirects(10).build(); httpPost.setConfig(requestConfig); // 增加头跳过 IP 认证
httpPost.addHeader("X_FORWARDED_FOR", "127.0.0.1"); httpPost.addHeader("x_forwarded_for",
"127.0.0.1"); // 设置请求体 postentity = new StringEntity( URLEncoder.encode(sendContent, "utf-8"));
postentity.setContentType("text/json;charset=utf-8"); postentity.setContentEncoding("utf-8");
httpPost.setEntity(postentity); System.out.println("url encode post enti:\n" + URLEncoder.encode(sendContent,
"utf-8")); CloseableHttpResponse result = httpClient.execute(httpPost);
// 请求发送成功, 并得到响应 if (result.getStatusLine().getStatusCode() == HttpStatus.SC_OK) {
cr.setCode(200); cr.setSuccess(true); respstr = EntityUtils.toString(result.getEntity(), "utf-8");
cr.setRespontext(respstr); } else { cr.setCode(result.getStatusLine().getStatusCode()); cr.setSuccess(false); } }
catch (Exception e) { log.error("接口调用失败", e); cr.setCode(500); } finally { try { // 释放链接
httpPost.releaseConnection(); } catch (Exception e2) { } } return cr; } public static CallResult
callWsAttachment(String url, String appkey, String appsecret, Object bizparams, Map<String, File>
attachements, String method) { CallResult cr = new CallResult(); cr.setSuccess(false); HttpEntity postentity =
null; String respstr = ""; List<NameValuePair> queryparams = null; HttpPost httpPost = null; try {
CloseableHttpClient httpClient = HttpClients.createDefault(); // 创建请求 URL String query = ""; String
sendContent = ""; sendContent = encodeJson(bizparams); System.out.println("\n send json post:\n" +
sendContent); // 控制参数 String sign = generalContentSign(sendContent, appsecret); queryparams = new
ArrayList<NameValuePair>(); queryparams.add(new BasicNameValuePair("method", method));
queryparams.add(new BasicNameValuePair("sign", sign)); queryparams.add(new
BasicNameValuePair("appKey", appkey)); UrlEncodedFormEntity queryBuilder = new UrlEncodedFormEntity(
queryparams); ByteArrayOutputStream bout = new ByteArrayOutputStream(); queryBuilder.writeTo(bout);
bout.flush(); query = new String(bout.toByteArray(), "utf-8"); bout.close(); if

```

```
(DEV_MODE) { url = url + "?" + query +
```

```
"&XDEBUG_SESSION_START=ECLIPSE_DBGP&KEY=15022624739804"; } else { url = url + "?" + query; }
log.info("\n\n\n 处理后的请求 url:" + url); httpPost = new HttpPost(url); // 设置请求性能参数 RequestConfig
requestConfig = RequestConfig.custom().setSocketTimeout(115000).setConnectTimeout(115000)
.setMaxRedirects(10).build(); httpPost.setConfig(requestConfig); // 增加头跳过 IP 认证
httpPost.addHeader("X_FORWARDED_FOR", "127.0.0.1"); httpPost.addHeader("x_forwarded_for",
"127.0.0.1"); // 设置请求体 // 使用多部件模式 StringBody business_content = new
StringBody(URLEncoder.encode(sendContent, "utf-8")); MultipartEntity reqEntity = new MultipartEntity();
reqEntity.addPart("business_content", business_content); // file1 为请求后台的 File upload; 属性 if (attachements
!= null && attachements.size() > 0) { Set<String> attachenames = attachements.keySet(); for (String attname :
attachenames) { System.out.println("att name:" + attname); File sendattch = attachements.get(attname); if
(sendattch.exists()) { reqEntity.addPart( attname, new FileBody(sendattch));
} } } httpPost.setEntity(reqEntity); System.out.println("url encode post enti:\n" +
URLEncoder.encode(sendContent, "utf-8")); CloseableHttpResponse result = httpClient.execute(httpPost); // 请
求发送成功, 并得到响应 if (result.getStatusLine().getStatusCode() == HttpStatus.SC_OK) { cr.setCode(200);
cr.setSuccess(true); respstr = EntityUtils.toString(result.getEntity(), "utf-8"); cr.setRespontext(respstr); } else {
cr.setCode(result.getStatusLine().getStatusCode()); cr.setSuccess(false); } } catch (Exception e) { log.error("接
口调用失败", e); cr.setCode(500); } finally { try { // 释放链接 httpPost.releaseConnection(); } catch (Exception e2)
{ } } return cr; } }
```

FAQ:

什么是 DES 加盐呢?

那什么是加盐呢? 当我第一次看到这个词的时候, 我想到了我妈做的饭, 因为我妈做饭一直都很淡 ==

回到咱们要讲的加盐(Salt)。其实加盐是为了应对这么一种情况: 如果两个人或多个人的密码相同, 那么通过相同的加密算法得到的是相同的结果。这样会造成哪些后果呢? 首先, 破解一个就有可能是相当于破一片密码。而且加入小明这个用户可以查看后台数据库, 那么如果他观察到小红这个用户的密码跟自己的密码是一样的(虽然都是密文), 那么, 也就代表他们两个人的密码是相同的。所以他就可以用小红的身份进行登录了。

其实, 我们只要稍微混淆一下就能防范住了, 这在加密术语中称为“加盐”。具体来说就是在原有材料(用户自定义密码)中加入其他成分(一般是用户自有且不变的因素), 以此来增加系统复杂度。当这种盐和用户密码结合后, 再通过摘要处理, 就能得到隐蔽性更强的摘要值。

什么是 BASE64 算法呢?

参考这篇文章 <https://www.cnblogs.com/xqxacm/p/4886299.html>

接口文档

启动触发任务

基本信息

接口名称:启动触发任务

接口地址

:http://send.salepush.cn/edmcloud/api/api_route.php?method=startTriggerMailTask&sign=xxx&apiKey=x xx

请求方法:POST

请求数据类型:JSON

响应类型:JSON

状态:有效

接口描述:
开启一个发送触发邮件的投递任务

请求头

| 参数名称 | 是否必须 | 默认值 | 描述 |
|-------|------|-----|----|
| metho | true | | |

请求参数

| 参数名称 | 是否必须 | 数据类型 | 默认值 | 描述 |
|--------------|-------|---------------|-----|---|
| taskname | true | string | | 任务名称随意写. |
| subject | true | string | | 投递到您客户邮箱中的邮件主题 |
| domainid | false | string | | 使用发信域域 ID,目前单域名模式此参数暂时不用 |
| recipients | true | string | | 收件人的邮件地址 |
| group | false | string | | 任务所属分组,用于对您的一批触发任务进行分组 |
| content_tpl | true | string | | 模板 ID,此 ID 您可以在用户界面的模板管理列表第一列获得,模板就是您发送出去邮件内容,注意: 此模板必须是审核通过的 |
| fromname | true | string | | 发送人邮件地址,用于您客户收到邮件后发件人一项显示内容 |
| reply | false | string | | 回复地址 |
| need_receipt | false | boolean | | 是否需要发送回执 |
| tpl_vars | false | array[object] | | 模板变量 |

响应参数

| 参数名称 | 是否必须 | 数据类型 | 描述 |
|------|------|------|----|
| | | | |

| | | | |
|---------|------|--------|--|
| code | true | number | 200 调用成功,500 调用失败,416 缺少必要参数(接口通用参数缺失),403 认证失败(签名和内容部匹配,或者使用过期秘钥) |
| retcode | true | string | 业务错误故障代码:1001 无效模板 ID,1002 投递失败,错误的收信地址或者无效的信件主题 |
| data | true | string | 您发送任务详细信息 |

通用接口参数说明

基本信息

接口名称:通用接口参数说明

接口地址:所有我们接口地址

请求方法:GET

请求数据类型:X-WWW-FORM-URLENCODED

响应类型:JSON

状态:有效

接口描述:

请求参数

| 参数名称 | 是否必须 | 数据类型 | 默认值 | 描述 |
|--------|------|--------|-------------------|----|
| method | true | string | 业务方法名,每个不同的业务操作不同 | |
| apiKey | true | string | 用于密钥对身份标识 | |
| sign | true | string | 接口内容签名 | |

示例数据

http://send.salepush.cn/edmcloud/api/api_route.php?method=startTriggerMailTask&sign=RUQKixm94oi

QzRBc%2Fy8Ed998uUsbG8D%2Buw9OpLrYd%2BBHQpxYKTJAMg%3D%3D&apiKey=test%40send.com