云邮 OpenAPI 接口使用说明

目录

[一、 Open API HTTP 接口简介 3](#_Toc3671)

[二、 Open API HTTP 接口说明 3](#_Toc30525)

[三、 HTTP 接口请求参数说明 3](#_Toc9706)

[1. 接口协议 3](#_Toc31349)

[2. 公共请求参数说明 4](#_Toc17680)

[3. 加密方式 4](#_Toc3773)

[四、 接口调用方式注意事项](#_Toc28329)

[五、 接口详情](#_Toc11201)

[1. addEmployee 5](#_Toc25300)

[2. updateEmployee 6](#_Toc9147)

[3. deleteEmployee 8](#_Toc23316)

[4. addBranch 9](#_Toc398)

[5. updateBranch 10](#_Toc14470)

[6. deleteBranch 11](#_Toc9370)

[7. syncAllOrg 12](#_Toc15576)

[8. Login 14](#_Toc28772)

[9. getFolderList 15](#_Toc27614)

[10. getMailList 16](#_Toc22085)

[11. getSendMailURL 17](#_Toc25341)

[12. checkEmailIsExist 18](#_Toc29114)

[13. verifyEmployee 19](#_Toc7366)

[14. getMailPageURL 19](#_Toc29707)

[15. getReadMailURL 19](#_Toc2668)

[16. getToSystemPageUrl 20](#_Toc17001)

[17. getAllBranch 20](#_Toc31236)

[18. getAllEmployee 22](#_Toc430)

[19. getEmployee 23](#_Toc14255)

[20. modifyEmployeeServiceStatus 25](#_Toc25953)

[21. searchEmployeesServiceStatus 26](#_Toc2330)

[22. searchMails 27](#_Toc950)

[23. getMail 28](#_Toc24057)

[24. sendMail 31](#_Toc16026)

[六、 错误码说明](#_Toc9578)

[七、 附录](#_Toc4619)

# 一、 Open API HTTP 接口简介

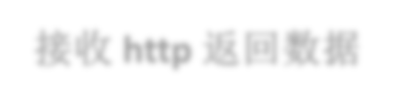
### Open API HTTP 接口（以下简称 HTTP 接口）是一个供程序员调用的、HTTP 形式的服务检索接口，向云邮箱用户开放。

目前 HTTP 接口可以服务于直接的 URl 访问，也可以服务那些非网页程序的调用，例如： C# 、C++、Java 等开发的应用程序，这些开发语言都能发送 HTTP 请求且能接收返回数据。HTTP 接口返回 json 形式的数据。

目前提供的功能不是很多、以后随着用户需求的收集和总结，会不断完善 HTTP 接口。

# 二、 Open API HTTP 接口说明

HTTP 接口使用分为 3 个步骤：第一步，获取 AppKey；第二步，拼写 HTTP 请求发送的 url， url 中附带上第一步获取的 AppKey 值；第三步，接收 HTTP 请求返回的 String 或者 Json 格式的数据，如下图示意：



**获取AppKey**

**拼写http 请求url & AppKey**

**接收http 返回数据**

获取 AppKey 说明：

用户可以直接在”域管理平台“后台的“域设置”获取。该 AppKey 主要目的是判断用户请求的 url 是否合法，且限制用户 1 天发送的 HTTP 请求数不能超过 1000 次。

# 三、 HTTP 接口请求参数说明

# 接口协议

参数以 HTTP 请求方式传递，格式示例：

[http://mail.xxx.com/domain/OnlineSearchServlet?requesttime=1402041034317&me](http://mail.xxx.com/domain/OnlineSearchServlet?requesttime=1402041034317&amp;me) thod=getMailsCount&email=a0\_1@t6000.35test.cn&param=paramValue&chksum=xxxxx &charset=utf-8

# 公共请求参数说明

|  |  |  |
| --- | --- | --- |
| 参数名 | 参数说明 | 参数设置 |
| requesttime | 请求时间, 13 位的 Long 类型值 | required |
| method | 对应服务的接口提供的功能 | required |
| email | email(英文且长度不大于 63):如：123@+  公司域名即可。[123@35.cn](mailto:123@35.cn) 等等 | required |
| param | 请求参数字符串（加密方式查看 1.3.3） | required |
| chksum | 为加密后的字符串 | required |
| charset | 默认为 utf-8 | required |

其中：required 为必须传递字段；optional 为可选传递字段；

# 加密方式

param 加密方式：

param = key1:base64(value1)|key2:base64(value2)|key3:... 如果无请求参数，则需要设置默认值，如下：

param = displayType:MA== chksum 加密方式：

是对 requesttime + method+ email + param + AppKey 进行 MD5 处理的字符串。

**四、 接口调用方式注意事项**

接口 1-7 为新接口，参数 json 格式放在请求体中然后调用方式 param 设置为""，接口 8-18

位旧接口，参数拼接设置到 param 中

**五、 接口详情**

接口详情中,将针对 method 值的不同,对请求参数进行详细描述，接口中提到的时间参数一致以 GMT+8 时间为准。

# addEmployee

简介：获取用户信息

请求参数描述：

json 格式，放在 http 请求体中

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 格式说明 | 参数说明 | 参数设置 |
| employeeId | String | 用户唯一 id | required |
| email | String | 用户 email，email 不允许重复 | required |
| name | String | 用户姓名 | required |
| password | String | 用户密码，没有传，则使用域的默认密码 | optional |
| branchIds | String | 用户所属的部门 id 列表，多个部门，之间  使用英文逗号分隔 | required |
| sortNum | Integer | 排序序号，从 1 开始，越小排在越前面。  如果不传该字段，默认是在最后。 | optional |
| isTop | Integer | 是否置顶，1 表示置顶，0 表示不置顶。默  认是 0 | optional |
| workNumber | String | 用户工号，必须唯一 | optional |
| gender | String | 性别，male 或者 female，male 表示男性，  female 表示女性 | optional |
| birthday | String | 出生日期，"yyyy-MM-dd"格式 | optional |
| mobile | String | 手机号码 | optional |
| [telephone](http://www.baidu.com/link?url=Vm4SJUTZgT3-ikySYLIcE662LLwq_sm8suzclGq4ql_78kEj1tx86zZ6ZB1kkk8JCRxnzufNzkJMec_BoI8Ws4t7J2yAxfrhWET8dsTj8TC) | String | 固定号码(工作号码) | optional |
| company | String | 公司名称 | optional |
| position | String | 职位 | optional |
| homefax | String | 家庭传真 | optional |
| workaddress | String | 通讯地址 | optional |
| qq | String | qq 号码 | optional |
| msn | String | msn 号码 | optional |

示例

{

"employeeId":" 297ed9df5a6e1ed7015a6edef05c0027", "email": "[zhangsan@test1.com](mailto:zhangsan@test1.com)",

"name": " 张 三 ", "password": "asdfg234#",

"branchIds": " 297ed9df58246b490158246b722a0002", "sortNum": "1",

"isTop": "0",

"workNumber": "000001", "gender":"male",

"birthday":"1990-06-20", "mobile": "15999888666",

"t[elephone](http://www.baidu.com/link?url=Vm4SJUTZgT3-ikySYLIcE662LLwq_sm8suzclGq4ql_78kEj1tx86zZ6ZB1kkk8JCRxnzufNzkJMec_BoI8Ws4t7J2yAxfrhWET8dsTj8TC)": "05922123123"， "company"：" 公 司 名 称 ", "position": " 经 理 ", "homefax": "05922123122",

"workaddress": "工作地址 1",

"qq": "122234556",

"msn": "22222222"

}

响应参数描述，采用 json 格式

|  |  |  |
| --- | --- | --- |
| 参数名 | 参数格式 | 参数说明 |
| errorCode | String | 错误码，可能的错误码：200，500，1000，1001，1002 |
| errorMsg | String | 错误消息 |

示例：

{

"errorCode":"200", "errorMsg":"success"

}

# updateEmployee

简介：更新用户

请求参数描述：

json 格式，放在 http 请求体中

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 格式说明 | 参数说明 | 参数设置 |
| employeeId | String | 用户唯一 id | required |
| name | String | 用户姓名 | optional |
| password | String | 用户密码，没有传，则使用域的默认密码 | optional |
| branchIds | String | 用户所属的部门 id 列表，多个部门，之间  使用英文逗号分隔 | optional |
| sortNum | Integer | 排序序号，从 1 开始，越小排在越前面。 | optional |
| isTop | Integer | 是否置顶，1 表示置顶，0 表示不置顶。 | optional |
| workNumber | String | 用户工号，必须唯一 | optional |
| gender | String | 性别，male 或者 female，male 表示男性，  female 表示女性 | optional |
| birthday | String | 出生日期，"yyyy-MM-dd"格式 | optional |
| mobile | String | 手机号码 | optional |
| [telephone](http://www.baidu.com/link?url=Vm4SJUTZgT3-ikySYLIcE662LLwq_sm8suzclGq4ql_78kEj1tx86zZ6ZB1kkk8JCRxnzufNzkJMec_BoI8Ws4t7J2yAxfrhWET8dsTj8TC) | String | 固定号码(工作号码) | optional |
| company | String | 公司名称 | optional |
| position | String | 职位 | optional |
| homefax | String | 家庭传真 | optional |
| workaddress | String | 通讯地址 | optional |
| qq | String | qq 号码 | optional |
| msn | String | msn 号码 | optional |

注意：更新接口，只更新传递过来的字段，没有传递过来的字段，默认和原来一样。如果需要清空某个属性，则该属性值为空字符串(注意：name,password，branchIds 不允许被清空)。

示例

{

"employeeId":" 297ed9df5a6e1ed7015a6edef05c0027", "name": "张三",

"password": "asdfg234#",

"branchIds": " 297ed9df58246b490158246b722a0002",

"sortNum": "2",

"isTop": "1",

"workNumber": "000001",

"gender":"male",

"birthday":"1990-06-20", "mobile": "15999888666",

"t[elephone](http://www.baidu.com/link?url=Vm4SJUTZgT3-ikySYLIcE662LLwq_sm8suzclGq4ql_78kEj1tx86zZ6ZB1kkk8JCRxnzufNzkJMec_BoI8Ws4t7J2yAxfrhWET8dsTj8TC)": "05922123123",

"company"：" 公 司 名 称 ",

"position": "经 理",

"homefax": "05922123122",

"workaddress": "工作地址 1",

"qq": "122234556",

"msn": "22222222"

}

响应参数描述，采用 json 格式

|  |  |  |
| --- | --- | --- |
| 参数名 | 参数格式 | 参数说明 |
| errorCode | String | 错误码，可能的错误码：200，500，1000，1002，1003 |
| errorMsg | String | 错误消息 |

示例：

{

"errorCode":"200", "errorMsg":"success"

}

# deleteEmployee

简介：删除用户

请求参数描述：

json 格式，放在 http 请求体中

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 格式说明 | 参数说明 | 参数设置 |

|  |  |  |  |
| --- | --- | --- | --- |
| employeeId | String | 用户唯一 id | required |

示例：

{

"employeeId":" 297ed9df5a6e1ed7015a6edef05c0027",

}

响应参数描述，采用 json 格式

|  |  |  |
| --- | --- | --- |
| 参数名 | 参数格式 | 参数说明 |
| errorCode | String | 错误码，可能的错误码：200，500，1000 |
| errorMsg | String | 错误消息 |

示例：

{

"errorCode":"200", "errorMsg":"success"

}

# addBranch

简介：新增部门

请求参数描述：

json 格式，放在 http 请求体中

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 格式说明 | 参数说明 | 参数设置 |
| branchId | String | 部门唯一 id | required |
| parentId | String | 父部门 id，空串表示一级部门 | required |
| branchName | String | 部门名称 | required |
| sortNum | Integer | 排序序号，从 1 开始，越小排在越前面。  如果不传该字段，默认是在最后。 | optional |

示例：

{"branchId":" 297ed9df58246b490158246b722a0002",

"parentId": "",

"branchName": "测试部门",

"sortNum": "1"

}

响应参数描述，采用 json 格式

|  |  |  |
| --- | --- | --- |
| 参数名 | 参数格式 | 参数说明 |
| errorCode | String | 错误码，可能的错误码：200，500，1000，1004 |
| errorMsg | String | 错误消息 |

示例：

{

"errorCode":"200", "errorMsg":"success"

}

# updateBranch

简介：更新部门

请求参数描述：

json 格式，放在 http 请求体中

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 格式说明 | 参数说明 | 参数设置 |
| branchId | String | 部门唯一 id | required |
| parentId | String | 父部门 id，空串表示一级部门 | optional |
| branchName | String | 部门名称 | optional |
| sortNum | Integer | 排序序号，从 1 开始，越小排在越前面。 | optional |

注意：更新接口，只更新传递过来的字段，没有传递过来的字段，默认和原来一样。示例：

{"branchId":" 297ed9df58246b490158246b722a0002",

"parentId": "", "branchName": "测试部门 2",

"sortNum": "2"

}

响应参数描述，采用 json 格式

|  |  |  |
| --- | --- | --- |
| 参数名 | 参数格式 | 参数说明 |
| errorCode | String | 错误码，可能的错误码：200，500，1000，1004，1005 |
| errorMsg | String | 错误消息 |

示例：

{

"errorCode":"200", "errorMsg":"success"

}

# deleteBranch

简介：删除部门

请求参数描述：

json 格式，放在 http 请求体中

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 格式说明 | 参数说明 | 参数设置 |
| branchId | String | 部门唯一 id | required |

示例：

{"branchId":" 297ed9df58246b490158246b722a0002"

}

响应参数描述，采用 json 格式

|  |  |  |
| --- | --- | --- |
| 参数名 | 参数格式 | 参数说明 |

|  |  |  |
| --- | --- | --- |
| errorCode | String | 错误码，可能的错误码：200，500，1000，1006,1007 |
| errorMsg | String | 错误消息 |

示例：

{"errorCode":"200", "errorMsg":"success"

}

# syncAllOrg

简介：全量同步组织架构

注意事项：全量同步时，如果是部门新增：父部门的数据要排在子部门的前面，不然会出现父部门还没创建的问题。

请求参数描述：

json 格式，放在 http 请求体中

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 格式说明 | 参数说明 | 参数设置 |
| branchInfo | Json | 部门信息 | required |
| employeeInfo | Json | 用户信息 | required |

branchInfo 描述，为 json 数组，数组中 json 的格式说明如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 格式说明 | 参数说明 | 参数设置 |
| branchId | String | 部门唯一 id | required |
| parentId | String | 父部门 id，空串表示一级部门 | required |
| branchName | String | 部门名称 | required |
| sortNum | Integer | 排序序号，从 1 开始，越小排在越前面。  如果不传该字段，默认是在最后。 | optional |

employeeInfo 描述，为 json 数组，数组中 json 的格式说明如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 格式说明 | 参数说明 | 参数设置 |
| employeeId | String | 用户唯一 id | required |

|  |  |  |  |
| --- | --- | --- | --- |
| email | String | 用户 email，email 不允许重复 | required |
| name | String | 用户姓名 | required |
| password | String | 用户密码，没有传，则使用域的默认密码 | optional |
| branchIds | String | 用户所属的部门 id 列表，多个部门，之间  使用英文逗号分隔 | required |
| sortNum | Integer | 排序序号，从 1 开始，越小排在越前面。  如果不传该字段，默认是在最后。 | optional |
| isTop | Integer | 是否置顶，1 表示置顶，0 表示不置顶。默  认是 0 | optional |
| workNumber | String | 用户工号，必须唯一 | optional |
| gender | String | 性别，male 或者 female，male 表示男性，  female 表示女性 | optional |
| birthday | String | 出生日期，"yyyy-MM-dd"格式 | optional |
| mobile | String | 手机号码 | optional |
| [telephone](http://www.baidu.com/link?url=Vm4SJUTZgT3-ikySYLIcE662LLwq_sm8suzclGq4ql_78kEj1tx86zZ6ZB1kkk8JCRxnzufNzkJMec_BoI8Ws4t7J2yAxfrhWET8dsTj8TC) | String | 固定号码(工作号码) | optional |
| company | String | 公司名称 | optional |
| position | String | 职位 | optional |
| homefax | String | 家庭传真 | optional |
| workaddress | String | 通讯地址 | optional |
| qq | String | qq 号码 | optional |
| msn | String | msn 号码 | optional |

例子：

{

"branchInfo":[

{"branchId":" 297ed9df58246b490158246b722a0002",

"parentId": "", "branchName": "测试部门", "sortNum": "1"

},{…}],

"employeeInfo":[{

"employeeId":" 297ed9df5a6e1ed7015a6edef05c0027", "email": "[zhangsan@test1.com](mailto:zhangsan@test1.com)",

"name": " 张 三 ", "password": "asdfg234#",

"branchIds": " 297ed9df58246b490158246b722a0002", "sortNum": "1",

"isTop": "0",

"workNumber": "000001", "gender":"female",

"birthday":"1990-06-20", "mobile": "15999888666",

"t[elephone](http://www.baidu.com/link?url=Vm4SJUTZgT3-ikySYLIcE662LLwq_sm8suzclGq4ql_78kEj1tx86zZ6ZB1kkk8JCRxnzufNzkJMec_BoI8Ws4t7J2yAxfrhWET8dsTj8TC)": "05922123123", "company"：" 公 司 名 称 ", "position": " 经 理 ", "homefax": "05922123122",

"workaddress": "工作地址 1",

"qq": "122234556",

"msn": "22222222"

}, {…}]

}

响应参数描述，采用 json 格式

|  |  |  |
| --- | --- | --- |
| 参数名 | 参数格式 | 参数说明 |
| errorCode | String | 错误码，可能的错误码：200，500，1000，1001，1002，1004 |
| errorMsg | String | 错误消息 |

示例：

{"errorCode":"200", "errorMsg":"success"

}

# Login

简介：获取接口调用的token。若需实现单点登录，详见接口：getMailPageURL。

请求参数描述：

无

响应参数描述：

|  |  |  |
| --- | --- | --- |
| 参数名 | 格式说明 | 参数说明 |

|  |  |  |
| --- | --- | --- |
| tokenString | String | 返回此次会话的 Token，有效期20分钟 |

示例：

[{"tokenString":"A4EBD24B4F706EC0E7E0DA034AA06635"}]

# getFolderList

简介：获取邮箱文件夹信息请求参数描述：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 格式说明 | 参数说明 | 参数设置 |
| tokenString | String | 登录成功返回的 Token | required |
| displayType | String | 需要显示的文件夹:  0,所有(默认); 1,自定义文件夹 | optional |

响应参数描述：

|  |  |  |
| --- | --- | --- |
| 参数名 | 参数格式 | 参数说明 |
| folderId | String | 文件夹 ID |
| folderName | String | 文件夹名称 |
| folderType | String | 0=系统 1=自定义 2=归档 3=其它邮箱 |
| orderValue | String | 排序值 |
| totalCount | String | 邮件总数 |
| unreadCount | String | 未读数 |

示 例 ：

[{"folderId":"Destroy","folderName":" 自摧毁

","folderType":"0","orderValue":"6","totalCount":"0","unreadCount":"0"},

{"folde rId":".Draft","folderName":" 草稿箱","folderType":"0","orderValue":"5","totalCount":"0","unreadCount":"0"},

{"folde rId":".Spam","folderName":" 垃圾箱","folderType":"0","orderValue":"4","totalCount":"0","unreadCount":"0"},

{"folde rId":".Trash","folderName":" 已删除","folderType":"0","orderValue":"3","totalCount":"0","unreadCount":"0"},

{"folde rId":".Sent","folderName":" 已发送

","folderType":"0","orderValue":"2","totalCount":"1","unreadCount":"0"},

{"folde rId":"inbox","folderName":" 收件箱","folderType":"0","orderValue":"1","totalCount":"2","unreadCount":"2"}]

# getMailList

简介：通过目录得到邮件列表请求参数描述：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 参数格式 | 参数说明 | 参数设置 |
| tokenString | String | 登录成功返回的 Token | required |
| folderId | String | 目录名(必填项) | required |
| state | int | 邮件状态（选填项）： 0=所有,1=已读,2=未读  没有传这个参数，默认是所有 | [optional](http://www.baidu.com/link?url=6goUVZHBE1SwAmHXNn2SXdF_ug_MRWttMFGVCixTDDwLx9cISkN-SNY07zUpnFeQkTYynNKDTWEK9zwLtvinqSgWSmHR5_a3dfHwXLMl5r3) |
| pageNo | int | 第几页(必填项，数值范围：>=1) | required |
| pageSize | int | 页大小(必填项，数值范围：>=1) | required |
| newWindow | String | 是否新窗口：1 是 | [optional](http://www.baidu.com/link?url=6goUVZHBE1SwAmHXNn2SXdF_ug_MRWttMFGVCixTDDwLx9cISkN-SNY07zUpnFeQkTYynNKDTWEK9zwLtvinqSgWSmHR5_a3dfHwXLMl5r3) |

响应参数描述，采用 json 格式

|  |  |  |
| --- | --- | --- |
| 参数名 | 参数格式 | 参数说明 |
| maidId | String | 邮件 ID |
| from | String | 发件人 |
| to | String | 收件人 |
| cc | String | 抄送人 |
| bcc | String | 密送人 |
| sendTime | String | 发送时间 |
| subject | String | 邮件主题 |
| read | int | 是否已读 0=未读 1=已读 |
| reply | String | 回复日期如(2010-05-13 12:00:00)，空表示未回复 |
| mailType | String | 邮件类型.0=普通邮件 1=会议邮件 2=自摧毁邮 3=  定时邮件 |
| folderId | String | 邮件夹ID |
| mailURL | String | 此封邮件的超链接，用于查看邮件内容 |
| plainText | String | 纯文本 |

示例：

[{

"mailId":"a",

"from":"b",

"to":"c",

"cc":"",

"bcc":"",

"sendTime":"2014-01-01 12:00:00",

"subject":"f",

"read":1,

"reply":"",

"mailType":"1",

"folderId":"inbox",

"mailURL":"[http://mail.sofia7.com/read](http://mail.sofia3.com/read) mail?EaQTYnHTgHEAW97VmDVoOaGd1XQAeUw5WwfNK8YZZqLfuLrvX0EELxUtFThrdUwYZ7C7K5 1/E8D4ZmB0mCvsNlrjCV2gqlfJ",

"plainText":"test"

}]

# getSendMailURL

简介：获取写信 URL 请求参数描述：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 参数格式 | 参数说明 | 参数设置 |
| tokenString | String | 登录成功返回的 Token | required |
| recv | String | 收件人，多个地址,按分号分隔 | [optional](http://www.baidu.com/link?url=6goUVZHBE1SwAmHXNn2SXdF_ug_MRWttMFGVCixTDDwLx9cISkN-SNY07zUpnFeQkTYynNKDTWEK9zwLtvinqSgWSmHR5_a3dfHwXLMl5r3) |
| newWindow | String | 是否新窗口：1 是 | [optional](http://www.baidu.com/link?url=6goUVZHBE1SwAmHXNn2SXdF_ug_MRWttMFGVCixTDDwLx9cISkN-SNY07zUpnFeQkTYynNKDTWEK9zwLtvinqSgWSmHR5_a3dfHwXLMl5r3) |
| mailId | String | 草稿邮件ID | [optional](http://www.baidu.com/link?url=6goUVZHBE1SwAmHXNn2SXdF_ug_MRWttMFGVCixTDDwLx9cISkN-SNY07zUpnFeQkTYynNKDTWEK9zwLtvinqSgWSmHR5_a3dfHwXLMl5r3) |

响应参数描述，采用 json 格式

|  |  |  |
| --- | --- | --- |
| 参数名 | 参数格式 | 参数说明 |
| sendMailURL | String | 获取写信 URL |

示例：

[{"sendMailURL":"http://mail.sofia7.com/sendmail?8Xl51jBDHgTL5Jdr/AaFoLS4eTRkc69kJRn2+o7wHRxx/iQ9f+70f8ounyxn2sQ0"}]

# checkEmailIsExist

简介：校验 Email 是否存在请求参数描述：

无

响应参数描述，采用 json 格式

|  |  |  |
| --- | --- | --- |
| 参数名 | 参数格式 | 参数说明 |
| emailIsExist | String | “1",存在；"0",不存在 |

示例：

[{"emailIsExist":"1"}]

# verifyEmployee

简介：账号密码校验

请求参数描述

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 参数格式 | 参数说明 | 参数设置 |
| password | String | 密码 | required |

响应参数描述：

|  |  |  |
| --- | --- | --- |
| 参数名 | 格式说明 | 参数说明 |
| result | String | 1，校验成功；0，校验失败 |

示例：

[{"result":"1"}]

# getMailPageURL

简介：获取邮箱首页的 URL，可以使用这个链接直接登录邮箱请求参数描述：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 参数格式 | 参数说明 | 参数设置 |
| tokenString | String | 登录成功返回的 Token | required |

响应参数描述，采用 json 格式

|  |  |  |
| --- | --- | --- |
| 参数名 | 参数格式 | 参数说明 |
| url | String | 获取 URL |

示例：

[{"url":"http://mail.sofia7.com/mailbox?2TEvm/GK/xLL5Jdr/AaFoNgQibjkO7Ip6bvVGBRvtf9It aALheBdTe7SFMnfqvbW"}]

# getReadMailURL

简介：获取某个邮件的详情的 URL 请求参数描述：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 参数格式 | 参数说明 | 参数设置 |
| tokenString | String | 登录成功返回的 Token | required |
| maidId | String | 邮件的 mailId | required |
| newWindow | String | 是否新窗口：1 是 | [optional](http://www.baidu.com/link?url=6goUVZHBE1SwAmHXNn2SXdF_ug_MRWttMFGVCixTDDwLx9cISkN-SNY07zUpnFeQkTYynNKDTWEK9zwLtvinqSgWSmHR5_a3dfHwXLMl5r3) |

响应参数描述，采用 json 格式

|  |  |  |
| --- | --- | --- |
| 参数名 | 参数格式 | 参数说明 |
| url | String | 获取 URL |

示例：

[{"url":"http://mail.sofia7.com/readmail?2TEvm/GK/xLL5Jdr/AaFoNgQibjkO7Ip6bvVGBRvtf9It

aALheBdTe7SFMnfqvbW"}]

# getToSystemPageUrl

简介：获取邮箱首页的 URL，可以使用这个链接直接登录邮箱请求参数描述：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 参数格式 | 参数说明 | 参数设置 |
| tokenString | String | 登录成功返回的 Token | required |
| toSystem | String | 签批：oa  邮局：mail | optional |

响应参数描述，采用 json 格式

|  |  |  |
| --- | --- | --- |
| 参数名 | 参数格式 | 参数说明 |
| url | String | 获取 URL |

示例：

[{"url":"http://mail.sofia7.com/mailbox?2TEvm/GK/xLL5Jdr/AaFoNgQibjkO7Ip6bvVGBRvtf9It aALheBdTe7SFMnfqvbW"}]

# getAllBranch

简介：获取组织结构

请求参数描述：

无

响应参数描述，采用 json 格式

json 格式

|  |  |  |
| --- | --- | --- |
| 参数名 | 参数格式 | 参数说明 |
| errorCode | String | 错误码，可能的错误码：200，500，2001 |
| errorMsg | String | 错误消息 |
| result | Json | 部门信息 |

result描述，为 json 数组，数组中 json 的格式说明如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 格式说明 | 参数说明 | 参数设置 |
| branchId | String | 部门唯一 id | required |
| parentId | String | 父部门 id，0表示一级部门 | required |
| branchName | String | 部门名称 | required |
| sortNum | Integer | 排序序号，从 1 开始，越小排在越前面。 | optional |

示例：

{

"result":[

{

"branchId":"ff8080815fa52d29016020046fb00c6f",

"branchName":"测试1",

"parentId":"0"

,"sortNum":3

},

{"branchId":"4028b88160bef9090160befa90330000",

"branchName":"测试2"

,"parentId":"ff8080815fa52d29016020046fb00c6f"

,"sortNum":1

},

{"branchId":"ff8080815fa52d290160068e98d50b44",

"branchName":"默认部门",

"parentId":"0",

"sortNum":2

}],

"errorCode":"200",

"errorMsg":"success"

}

# getAllEmployee

简介：获取所有用户

请求参数描述：

无

响应参数描述，采用 json 格式

json 格式

|  |  |  |
| --- | --- | --- |
| 参数名 | 参数格式 | 参数说明 |
| errorCode | String | 错误码，可能的错误码：200，500，2001 |
| errorMsg | String | 错误消息 |
| result | Json | 用户信息 |

result描述，为json 数组，数组中json的格式说明如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 格式说明 | 参数说明 | 参数设置 |
| employeeId | String | 用户唯一 id | required |
| email | String | 用户 email，email 不重复 | required |
| name | String | 用户姓名 | required |
| branchIds | String | 用户所属的部门 id 列表，多个部门，之间  使用英文逗号分隔 | required |
| sortNum | Integer | 排序序号，从 1 开始，越小排在越前面。 | optional |
| serviceInfos | String | 用户已开通服务，之间使用英文逗号分隔（mailserviceid为邮局） | optional |
| workNumber | String | 用户工号，唯一 | optional |

示例：

{

"result":[

{

"branchIds":"ff8080815fa52d29016020046fb00c6f,ff8080815fa52d290160068e98d50b44",

"email":"test1@sofia7.com",

"employeeId":"ff80808167ee44fb0167eecf2be40158",

"name":"test1",

"serviceInfos":"mailserviceid",

"sortNum":1,

"workNumber":"1"

},

{

"branchIds":"0",

"email":"test2@sofia7.com",

"employeeId":"ff80808167ca454f0167ea7fc6a001e2",

"name":"test2",

"serviceInfos":"mailserviceid",

"sortNum":2,

"workNumber":"2"

}

],

"errorCode":"200",

"errorMsg":"success"

}

# getEmployee

简介：获取用户信息

请求参数描述：

无

响应参数描述，采用 json 格式

json 格式

|  |  |  |
| --- | --- | --- |
| 参数名 | 参数格式 | 参数说明 |
| errorCode | String | 错误码，可能的错误码：200，500，1003，2001 |
| errorMsg | String | 错误消息 |
| result | Json | 用户信息 |

result描述，为json ，格式说明如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 格式说明 | 参数说明 | 参数设置 |
| employeeId | String | 用户唯一 id | required |
| email | String | 用户 email，email 不允许重复 | required |
| name | String | 用户姓名 | required |
| branchIds | String | 用户所属的部门id，多个部门，之间使用英文逗号分隔 | required |
| sortNum | Integer | 排序序号，从 1 开始，越小排在越前面。  如果不传该字段，默认是在最后。 | optional |
| workNumber | String | 用户工号，唯一 | optional |
| gender | String | 性别，male 或者 female，male 表示男性，  female 表示女性 | optional |
| birthday | String | 出生日期，"yyyy-MM-dd"格式 | optional |
| mobile | String | 手机号码 | optional |
| [telephone](http://www.baidu.com/link?url=Vm4SJUTZgT3-ikySYLIcE662LLwq_sm8suzclGq4ql_78kEj1tx86zZ6ZB1kkk8JCRxnzufNzkJMec_BoI8Ws4t7J2yAxfrhWET8dsTj8TC) | String | 固定号码(工作号码) | optional |
| company | String | 公司名称 | optional |
| position | String | 职位 | optional |
| homefax | String | 家庭传真 | optional |
| workaddress | String | 通讯地址 | optional |
| qq | String | qq 号码 | optional |
| msn | String | msn 号码 | optional |
| branchList | Json | 用户所属部门列表 | optional |
| serviceInfos | String | 用户已开通服务，之间使用英文逗号分隔（mailserviceid为邮局） | optional |

branchList描述，为json数组 ，数组中 json 的格式说明如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 格式说明 | 参数说明 | 参数设置 |
| branchId | String | 部门id | required |
| branchName | String | 部门名称 | required |
| position | String | 岗位，多个岗位，之间使用英文逗号分隔 | optional |

示例：

{

"result":{

"birthday":"2019-01-09",

"branchIds":"ff8080815fa52d290160068e98d50b44,ff808081658f5b080165adead64400bd",

"branchList":[

{

"branchId":"ff8080815fa52d290160068e98d50b44",

"branchName":"默认部门",

"position":"java工程师"

},

{

"branchId":"ff808081658f5b080165adead64400bd",

"branchName":"test2"

}

],

"company":"公司名",

"email":"test@sofia7.com",

"employeeId":"ff808081611d4a020161647047c4128c",

"gender":"male",

"homefax":"",

"mobile":"111111",

"msn":"",

"name":"test",

"position":"职位",

"qq":"",

"serviceInfos":"mailserviceid",

"sortNum":1,

"telephone":"222222",

"workNumber":"1",

"workaddress":""

},

"errorCode":"200",

"errorMsg":"success"

}

# modifyEmployeeServiceStatus

简介：修改用户服务状态

请求参数描述：

json 格式，放在 http 请求体中

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 参数格式 | 参数说明 | 参数设置 |
| list | String | Json数组 | required |

list描述，为 json 数组，数组中 json 的格式说明如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 格式说明 | 参数说明 | 参数设置 |
| email | String | 用户 email | required |
| service | String | 服务类型：mail | required |
| status | String | 状态：1开启，0禁用，-1删除 | required |

响应参数描述，采用 json 格式

|  |  |  |
| --- | --- | --- |
| 参数名 | 参数格式 | 参数说明 |
| errorCode | String | 错误码 |
| errorMsg | String | 错误消息 |
| result | Json | Json数组 |

result描述，为json 数组，数组中json的格式说明如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 格式说明 | 参数说明 | 参数设置 |
| email | String | 用户 email | required |
| result | String | 1成功，0失败 | required |
| msg | String | 失败原因 | optional |

# searchEmployeesServiceStatus

简介：搜索用户服务状态

请求参数描述：

json 格式，放在 http 请求体中（不能全为空，“并”关系查询）

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 格式说明 | 参数说明 | 参数设置 |
| name | String | 姓名 | optional |
| company | String | 公司名 | optional |
| branch | String | 部门名称 | optional |
| email | String | 用户email | optional |
| number | String | 用户工号 | optional |

响应参数描述，采用 json 格式

|  |  |  |
| --- | --- | --- |
| 参数名 | 参数格式 | 参数说明 |
| errorCode | String | 错误码 |
| errorMsg | String | 错误消息 |
| result | Json | Json数组 |

result描述，为json 数组，数组中json的格式说明如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 格式说明 | 参数说明 | 参数设置 |
| email | String | 用户 email | required |
| name | String | 姓名 | required |
| company | String | 公司名 | optional |
| branch | String | 部门名称 | optional |
| number | String | 用户工号 | optional |
| status | String | 状态：1开启，0禁用，-1无服务 | optional |
| time | String | 修改时间（开启时间、禁用时间，无服务时为空） | optional |

# searchMails

简介：搜索邮件

请求参数描述：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 参数格式 | 参数说明 | 参数设置 |
| tokenString | String | 登录成功返回的 Token | required |
| startTime | String | 发送开始时间  （格式：yyyy-MM-dd HH:mm:ss） | required |
| endTime | String | 发送结束时间  （格式：yyyy-MM-dd HH:mm:ss） | required |
| pageNo | int | 第几页(必填项，数值范围：>=1) | required |
| pageSize | int | 页大小(必填项，数值范围：1-100) | required |
| newWindow | String | 是否新窗口：1 是 | [optional](http://www.baidu.com/link?url=6goUVZHBE1SwAmHXNn2SXdF_ug_MRWttMFGVCixTDDwLx9cISkN-SNY07zUpnFeQkTYynNKDTWEK9zwLtvinqSgWSmHR5_a3dfHwXLMl5r3) |

响应参数描述，采用 json 格式

|  |  |  |
| --- | --- | --- |
| 参数名 | 参数格式 | 参数说明 |
| maidId | String | 邮件 ID |
| from | String | 发件人 |
| to | String | 收件人 |
| cc | String | 抄送人 |
| bcc | String | 密送人 |
| sendTime | String | 发送时间 |
| subject | String | 邮件主题 |
| read | int | 是否已读 0=未读 1=已读 |
| reply | String | 回复日期如(2010-05-13 12:00:00)，空表示未回复 |
| mailType | String | 邮件类型.0=普通邮件 1=会议邮件 2=自摧毁邮 3=  定时邮件 |
| folderId | String | 邮件夹ID |
| mailURL | String | 此封邮件的超链接，用于查看邮件内容 |
| plainText | String | 纯文本 |

示例：

{

"bcc": "",

"cc": "",

"folderId": "inbox",

"from": "test <test@sofia7.com>",

"maidId": "5f2b9ecb0cf29a36a44171cd",

"mailType": 0,

"mailURL": "http://mail.sofia7.com/readmail?EaQTYnHTgHGAfhqqwkHAVQ0T1HJfNXVrkk06yio3MNsrcLZh5zpvrgjrRgnEVHhWkEZILB3KfVuLiUTDTLeNB5A3D1rvY/94VSscDAapTmY=",

"read": 1,

"reply": "",

"sendTime": "2020-08-06 14:10:19",

"subject": "fix\_dp",

"to": "test <test@sofia7.com>",

"plainText": "test"

}

# getMail

简介：通过ID获取邮件

请求参数描述：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 参数格式 | 参数说明 | 参数设置 |
| tokenString | String | 登录成功返回的 Token | required |
| mailId | String | 邮件ID | required |
| newWindow | String | 是否新窗口：1 是 | [optional](http://www.baidu.com/link?url=6goUVZHBE1SwAmHXNn2SXdF_ug_MRWttMFGVCixTDDwLx9cISkN-SNY07zUpnFeQkTYynNKDTWEK9zwLtvinqSgWSmHR5_a3dfHwXLMl5r3) |

响应参数描述，采用 json 格式

|  |  |  |
| --- | --- | --- |
| 参数名 | 参数格式 | 参数说明 |
| maidId | String | 邮件 ID |
| from | String | 发件人 |
| to | String | 收件人 |
| cc | String | 抄送人 |
| bcc | String | 密送人 |
| sendTime | String | 发送时间 |
| subject | String | 邮件主题 |
| read | int | 是否已读 0=未读 1=已读 |
| reply | String | 回复日期如(2010-05-13 12:00:00)，空表示未回复 |
| mailType | String | 邮件类型.0=普通邮件 1=会议邮件 2=自摧毁邮 3=  定时邮件 |
| folderId | String | 邮件夹ID |
| mailURL | String | 此封邮件的超链接，用于查看邮件内容 |
| plainText | String | 纯文本 |
| hyperText | String | 富文本 |
| calendarText | String | 会议正文 |
| mailSize | long | 邮件大小 |
| attachmentCount | int | 附件数量 |
| attachmentSize | long | 附件大小 |
| attachmentList | JSON | 邮件附件 |

attachmentList描述，为json 数组，数组中json的格式说明如下：

|  |  |  |
| --- | --- | --- |
| 参数名 | 格式说明 | 参数说明 |
| name | String | 附件名称 |
| size | long | 附件大小 |
| url | String | 附件链接 |

示例：

{

"attachmentCount": 1,

"attachmentList": [{

"name": "test.txt",

"size": 8,

"url": "http://mail.sofia7.com/servlet/attachmentServlet?type=1&user=test@sofia7.com&mailId=5f30a8750cf284b8370d11fe&attachId=5f30a8750cf284b8370d11fe\_inbox\_0"

}],

"attachmentSize": 8,

"bcc": "",

"calendarText": "",

"cc": "",

"folderId": "inbox",

"from": "test <test@sofia7.com>",

"hyperText": "<div><div style=\"font-family:SimSun;font-size:18px;\"><div>测试<br><\/div><\/div> <\/div>",

"maidId": "5f30a8750cf284b8370d11fe",

"mailSize": 1198,

"mailType": 0,

"mailURL": "http://mail.sofia7.com/readmail?EaQTYnHTgHGAfhqqwkHAVQ0T1HJfNXVrkk06yio3MNvf25TOd+9EvW/OARUzHeyTnTBbdShVKlqjxV4P82w7RZvd67kysfS7UfrhBdWWTCs=",

"plainText": "测试\n\n\n \n",

"read": 1,

"reply": "",

"sendTime": "2020-08-10 09:52:52",

"subject": "test",

"to": "test <test@sofia7.com>"

}

# sendMail

简介：发送邮件

请求参数描述：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 参数格式 | 参数说明 | 参数设置 |
| tokenString | String | 登录成功返回的 Token | required |
| reMailId | String | 原邮件ID | [optional](http://www.baidu.com/link?url=6goUVZHBE1SwAmHXNn2SXdF_ug_MRWttMFGVCixTDDwLx9cISkN-SNY07zUpnFeQkTYynNKDTWEK9zwLtvinqSgWSmHR5_a3dfHwXLMl5r3) |
| to | String | 收件人 | required |
| subject | String | 邮件主题 | [optional](http://www.baidu.com/link?url=6goUVZHBE1SwAmHXNn2SXdF_ug_MRWttMFGVCixTDDwLx9cISkN-SNY07zUpnFeQkTYynNKDTWEK9zwLtvinqSgWSmHR5_a3dfHwXLMl5r3) |
| plainText | String | 纯文本 | required |
| type | String | 类型：save保存草稿；reply答复邮件 | [optional](http://www.baidu.com/link?url=6goUVZHBE1SwAmHXNn2SXdF_ug_MRWttMFGVCixTDDwLx9cISkN-SNY07zUpnFeQkTYynNKDTWEK9zwLtvinqSgWSmHR5_a3dfHwXLMl5r3) |

响应参数描述，采用 json 格式

|  |  |  |
| --- | --- | --- |
| 参数名 | 参数格式 | 参数说明 |
| errorCode | String | 错误码，可能的错误码：200,207,209,304 |
| errorMsg | String | 错误消息 |
| mailId | String | type为save时返回 |

示例：

{

"errorCode":"200",

"errorMsg":"success",

"maidId": "5f30a8750cf284b8370d11fe"

}

**六、 错误码说明**

|  |  |  |
| --- | --- | --- |
| 错误码 | 说明 | 备注 |
| 200 | 成功 |  |
| 500 | 系统异常 |  |
| 1000 | 必填字段为空 |  |
| 1001 | 用户 email 重复 |  |
| 1002 | 日期格式错误 |  |
| 1003 | 成员 id 不存在 |  |
| 1004 | 存在同一级相同名称的部门 |  |
| 1005 | 更新的部门不存在 |  |
| 1006 | 部门下还有子部门，不能进行删除 |  |
| 1007 | 部门下还有用户，不能进行删除 |  |
| 407080 | 新建部门失败，输入层级参数过大 |  |
| 407081 | 新建部门失败，公司 ID 不存在 |  |
| 407082 | 新建部门失败，负责人集合错误 |  |
| 407083 | 新建部门失败，父级部门 ID 不存在 |  |
| 407084 | 新建部门失败，层级超过上限 |  |
| 407085 | 新建部门失败，新建部门已存在 |  |
| 407086 | 新建部门失败 |  |
| 11001 | 入参数据类型错误 |  |
| 11002 | 父部门在 eis 系统中不存在 |  |
| 11003 | 获取公司信息失败 |  |
| 11100 | 要修改的部门在 eis 系统中不存在 |  |
| 407145 | 移动部门失败，新父级部门中出现同名 |  |
| 407143 | 移动部门失败，新的父级 ID 指向错误 |  |
| 407144 | 移动部门失败，累加层级关系过多 |  |
| 407140 | 移动部门失败，输入层级参数过大 |  |
| 407102 | 更新部门失败，新的部门名称已存在 |  |
| 1101 | 要删除的部门在 eis 系统中不存在 |  |
| 407090 | 删除部门失败，包含子部门、雇员或岗位 |  |
| 1201 | branchInfoList json 格式不正确 |  |
| 1202 | employeeInfoList json 格式不正  确 |  |
| 1203 | branchInfoList 数据存在必填项  没有值 |  |
| 1204 | employeeInfoList 数据存在必填  项没有值 |  |
| 1205 | branchInfo json 格式不正确 |  |
| 2000 | 域名不存在 |  |
| 2001 | 获取公司信息失败 |  |
| 498010 | 获取 eis 映射成员信息失败 |  |
| 498013 | 添加成员失败 |  |
| 498014 | 更新成员失败 |  |
| 498015 | 删除成员失败 |  |
| 2004 | Json 串为空 |  |
| 2010 | 更新成员的部门 id 不存在 |  |
| 2005 | 获取 eis 映射部门 id 失败 |  |
| 2007 | Json 中 branch 为空 |  |
| 2008 | 添加的成员 email 已经存在 |  |
| 2009 | 添加的成员用户 id 已经存在 |  |
| 2011 | 删除时请求参数中 email 为空 |  |
| 2012 | 创建用户时排序不能为 0 |  |
| 207 | 接收人地址不正确 |  |
| 209 | 邮箱空间满 |  |
| 304 | 邮件发送失败 |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**七、 附录**

本章节主要为开发人员提供针对以上接口的实现示例，方便开发人员进行开发。以下提供的是面向JAVA 语言的。

* 1. createEmployee

暂不提供，后续更新。

* 1. Login

**static void** login(String urlStr, String email) { System.*out*.println("Start testing HTTP client!"); **try** {

String charset = "utf-8"; String method = "login";

StringBuffer sb = **new** StringBuffer();

String param = sb.toString();

StringBuffer str = **new** StringBuffer(); str.append(*requestTime*); str.append(method); str.append(email);

str.append(param); str.append(*appKey*);

String chksum = DigestUtils.*md5Hex*(str.toString()).toLowerCase();

Map<String, String> paramMap = **new** HashMap<String, String>(); paramMap.put("requesttime", URLEncoder.*encode*(*requestTime*, "utf-8")); paramMap.put("method", URLEncoder.*encode*(method, "utf-8")); paramMap.put("email", URLEncoder.*encode*(email, "utf-8")); paramMap.put("chksum", URLEncoder.*encode*(chksum, "utf-8")); paramMap.put("param", URLEncoder.*encode*(param, "utf-8"));

paramMap.put("charset", URLEncoder.*encode*(charset, "utf-8")); HttpRequester request = **new** HttpRequester();

HttpRespons hr = request.sendPost(urlStr, paramMap);

System.*out*.println("Response: " + hr.getContent()); System.*out*.println("End of testing HTTP client!");

} **catch** (Exception e) { System.*out*.println(e.toString());

}

}

* 1. getFolderList

**static void** getFolderList(String urlStr, String email) { System.*out*.println("Start testing HTTP client!"); **try** {

BASE64Encoder be = **new** BASE64Encoder(); String charset = "utf-8";

String method = "getFolderList";

String tokenString = be.encode(*token*.getBytes());;

String displayType = be.encode("0".getBytes());//需要显示的文件夹: 0,所有; 1,自定义文件夹

StringBuffer sb = **new** StringBuffer(); sb.append("tokenString:"); sb.append(tokenString);

sb.append("|displayType:");//需要显示的文件夹:0,所有(默认); 1,自定义文件夹sb.append(displayType);

String param = sb.toString();

StringBuffer str = **new** StringBuffer(); str.append(*requestTime*); str.append(method); str.append(email);

str.append(param); str.append(*appKey*);

String chksum = DigestUtils.*md5Hex*(str.toString()).toLowerCase();

Map<String, String> paramMap = **new** HashMap<String, String>(); paramMap.put("requesttime", URLEncoder.*encode*(*requestTime*, "utf-8")); paramMap.put("method", URLEncoder.*encode*(method, "utf-8")); paramMap.put("email", URLEncoder.*encode*(email, "utf-8")); paramMap.put("chksum", URLEncoder.*encode*(chksum, "utf-8")); paramMap.put("param", URLEncoder.*encode*(param, "utf-8"));

paramMap.put("charset", URLEncoder.*encode*(charset, "utf-8")); HttpRequester request = **new** HttpRequester();

HttpRespons hr = request.sendPost(urlStr, paramMap);

System.*out*.println("Response: " + hr.getContent()); System.*out*.println("End of testing HTTP client!");

} **catch** (Exception e) { System.*out*.println(e.toString());

}

}

## getMailList

**static void** getMailList(String urlStr, String email) {

#### try {

BASE64Encoder be = **new** BASE64Encoder(); String charset = "utf-8";

String method = "getMailList";

String tokenString = be.encode(*token*.getBytes());; String folderId = be.encode("inbox".getBytes()); String pageNo = be.encode("1".getBytes());

String pageSize = be.encode("10".getBytes()); StringBuffer sb = **new** StringBuffer(); sb.append("tokenString:"); sb.append(tokenString); sb.append("|folderId:"); sb.append(folderId);

sb.append("|pageNo:");//第几页(必填项，数值范围：>=1) sb.append(pageNo);

sb.append("|pageSize:");//页大小(必填项，数值范围：>=1) sb.append(pageSize);

String param = sb.toString(); StringBuffer str = **new** StringBuffer(); str.append(*requestTime*); str.append(method); str.append(email);

str.append(param); str.append(*appKey*);

String chksum = DigestUtils.*md5Hex*(str.toString()).toLowerCase();

Map<String, String> paramMap = **new** HashMap<String, String>(); paramMap.put("requesttime", URLEncoder.*encode*(*requestTime*, "utf-8")); paramMap.put("method", URLEncoder.*encode*(method, "utf-8")); paramMap.put("email", URLEncoder.*encode*(email, "utf-8")); paramMap.put("chksum", URLEncoder.*encode*(chksum, "utf-8")); paramMap.put("param", URLEncoder.*encode*(param, "utf-8"));

paramMap.put("charset", URLEncoder.*encode*(charset, "utf-8")); HttpRequester request = **new** HttpRequester();

HttpRespons hr = request.sendPost(urlStr, paramMap); System.*out*.println("Response: " + hr.getContent());

} **catch** (Exception e) {

System.*out*.println(e.toString());

}

}

## getSendMailURL

**static void** getSendMailURL(String urlStr, String email) {

#### try {

BASE64Encoder be = **new** BASE64Encoder(); String charset = "utf-8";

String method = "getSendMailURL";

String tokenString = be.encode(token.getBytes());

StringBuffer sb = new StringBuffer();

sb.append("tokenString:"); sb.append(tokenString); String param = sb.toString();

StringBuffer str = **new** StringBuffer(); str.append(*requestTime*); str.append(method); str.append(email);

str.append(param); str.append(*appKey*);

String chksum = DigestUtils.*md5Hex*(str.toString()).toLowerCase();

System.*out*.println("chksum:" + chksum); System.*out*.println("param:" + param);

System.*out*.println(URLEncoder.*encode*(*requestTime*, "utf-8"));

Map<String, String> paramMap = **new** HashMap<String, String>(); paramMap.put("requesttime", URLEncoder.*encode*(*requestTime*, "utf-8")); paramMap.put("method", URLEncoder.*encode*(method, "utf-8")); paramMap.put("email", URLEncoder.*encode*(email, "utf-8")); paramMap.put("chksum", URLEncoder.*encode*(chksum, "utf-8")); paramMap.put("param", URLEncoder.*encode*(param, "utf-8"));

paramMap.put("charset", URLEncoder.*encode*(charset, "utf-8")); HttpRequester request = **new** HttpRequester();

HttpRespons hr = request.sendPost(urlStr, paramMap);

System.*out*.println("Response: " + hr.getContent());

} **catch** (Exception e) { System.*out*.println(e.toString());

}

}

## checkEmailIsExist

**static void** checkEmailIsExist(String urlStr, String email) {

#### try {

BASE64Encoder be = **new** BASE64Encoder(); String charset = "utf-8";

String method = "checkEmailIsExist";

String tokenString = be.encode(token.getBytes());

StringBuffer sb = new StringBuffer();

sb.append("tokenString:");

sb.append(tokenString);

String param = sb.toString();

StringBuffer str = **new** StringBuffer(); str.append(*requestTime*); str.append(method); str.append(email);

str.append(param); str.append(*appKey*);

String chksum = DigestUtils.*md5Hex*(str.toString()).toLowerCase();

Map<String, String> paramMap = **new** HashMap<String, String>(); paramMap.put("requesttime", URLEncoder.*encode*(*requestTime*, "utf-8")); paramMap.put("method", URLEncoder.*encode*(method, "utf-8")); paramMap.put("email", URLEncoder.*encode*(email, "utf-8")); paramMap.put("chksum", URLEncoder.*encode*(chksum, "utf-8")); paramMap.put("param", URLEncoder.*encode*(param, "utf-8"));

paramMap.put("charset", URLEncoder.*encode*(charset, "utf-8")); HttpRequester request = **new** HttpRequester();

HttpRespons hr = request.sendPost(urlStr, paramMap);

System.*out*.println("Response: " + hr.getContent());

} **catch** (Exception e) {

System.*out*.println(e.toString());

}

}

* 1. getMailPageURL

**static void** getMailPageURL(String urlStr, String email) { System.*out*.println("Start testing HTTP client!");

#### try {

BASE64Encoder be = **new** BASE64Encoder(); String charset = "utf-8";

String method = "getMailPageURL";

String tokenString = be.encode(*token*.getBytes());;

StringBuffer sb = **new** StringBuffer(); sb.append("tokenString:"); sb.append(tokenString);

String param = sb.toString();

StringBuffer str = **new** StringBuffer(); str.append(*requestTime*); str.append(method); str.append(email);

str.append(param); str.append(*appKey*);

String chksum = DigestUtils.*md5Hex*(str.toString()).toLowerCase();

Map<String, String> paramMap = **new** HashMap<String, String>(); paramMap.put("requesttime", URLEncoder.*encode*(*requestTime*, "utf-8")); paramMap.put("method", URLEncoder.*encode*(method, "utf-8")); paramMap.put("email", URLEncoder.*encode*(email, "utf-8")); paramMap.put("chksum", URLEncoder.*encode*(chksum, "utf-8")); paramMap.put("param", URLEncoder.*encode*(param, "utf-8"));

paramMap.put("charset", URLEncoder.*encode*(charset, "utf-8")); HttpRequester request = **new** HttpRequester();

HttpRespons hr = request.sendPost(urlStr, paramMap); System.*out*.println("Response: " + hr.getContent());

} **catch** (Exception e) {

System.*out*.println(e.toString());

}

}