MailBrid API接口

**目 录**

一、 邮件发送接口 2

1)接口相关说明 2

2） 示例代码：C# 3

3)示例代码：java 5

3） 示例代码:php 7

二、 获取发信通道接口 8

1)接口相关说明 8

三、 获取统计报表接口 9

1)接口相关说明 9

2) 返回数据格式说明 9

#### 邮件发送接口

##### 1)接口相关说明

接口地址:https://api.mailbrid.com/apiv1/send

数据提交方式:post

参数说明

|  |  |  |
| --- | --- | --- |
| **参数名称** | **类型** | **备注** |
| accessKeyID | string | 可在平台中获取 |
| accessToken | string | 产生方法:accessToken=md5(accessKeyID+accessKey+注册时的email+注册时的手机号码+sendtime参数) |
| taskname | string | 任务名称 |
| title | string | 邮件标题 |
| summary | string | 邮件摘要,可为空 |
| content | string | 邮件内容，支持html ,可插入变量，形如:{变量名}各变量名称如下：{EAddr}：电子邮箱{UserName}：姓名{NickName}：昵称{Sex}：性别{Gender}：称呼{Birthday}：生日{Old}：年龄{Mobile}：手机号码{Addr}：地址 |
| tomail | string | 收件箱，多个收件箱用“;”分开，支持变量,变量之间用“，”分隔，形如：邮件地址,姓名,昵称,性别,称呼,生日,年龄,手机号码,联系地址如: b@abc.com,王丽丽,小顽童,女,女士,1996-5-8,22岁,13913212342,北京海淀区;c@abc.com,刘三，小刘，男，先生，1998-10-10，25岁，18967887809，宁波北仑区 |
| fromdns | string | 发信域名，必须在平台内配置并验证通过的域名,形如：noreply@message.mailbrid.net |
| reply | string | 回复地址，可为空 |
| sendername | string | 发件人 |
| pipeid | string | 发信通道id,可调用接口getpipe 获取 |
| sendtime | datetime | 发送时间，产生accessToken用 |

##### **示例代码：C#**

using System.Security.Cryptography;

using System.Text;

using System.Net.Http;

using System.Collections.Generic;

 void send()

{

 String accessKeyID = "你的accessKeyID";

 string accessKey = "你的accessKey";

 string loginname = "你注册时的邮箱地址";

 string mobile = "你注册时的手机号码";

 string timestr = DateTime.Now.ToString ("yyyy-MM-dd HH:mm:ss");

 string accessToken = Md5(accessKeyID + accessKey + loginname + mobile + timestr );

 String url = "https://api.mailbrid.com/apiv1/send";

 HttpClient client = null;

 HttpResponseMessage response = null;

 try

 {

 client = new HttpClient();

 List<KeyValuePair<String, String>> paramList = new List<KeyValuePair<String, String>>();

 paramList.Add(new KeyValuePair<string, string>("accessKeyID", accessKeyID));

 paramList.Add(new KeyValuePair<string, string>("accessToken", accessToken));

 paramList.Add(new KeyValuePair<string, string>("taskname", "任务名称"));

 paramList.Add(new KeyValuePair<string, string>("title", "邮件标题"));

 paramList.Add(new KeyValuePair<string, string>("summary", ""));

 paramList.Add(new KeyValuePair<string, string>("content", "邮件内容");

 paramList.Add(new KeyValuePair<string, string>("tomail", "收件地址列表"));

 paramList.Add(new KeyValuePair<string, string>("fromdns", "发件地址"));

 paramList.Add(new KeyValuePair<string, string>("reply", "回复地址"));

paramList.Add(new KeyValuePair<string, string>("sendername", "发件人名称"));

 paramList.Add(new KeyValuePair<string, string>("pipeid", "通道id"));

 paramList.Add(new KeyValuePair<string, string>("sendtime", timestr));

 response = client.PostAsync(url, new FormUrlEncodedContent(paramList)).Result;

 String result = response.Content.ReadAsStringAsync().Result;

 Console.WriteLine("result:{0}", result);

 }

 catch (Exception e)

 {

 Console.WriteLine("\nException Caught!");

 Console.WriteLine("Message :{0} ", e.Message);

 }

 finally

 {

 if (null != client)

 {

 client.Dispose();

 }

 }

}

 string Md5(string value)

{

 byte[] bytes;

 using (var md5 = MD5.Create())

 {

 bytes = md5.ComputeHash(Encoding.UTF8.GetBytes(value));

 }

 var result = new StringBuilder();

 foreach (byte t in bytes)

 {

 result.Append(t.ToString("X2"));

 }

 return result.ToString().ToLower();

}

**3)示例代码：java**

import java.io.IOException;

 import java.util.ArrayList;

 import java.util.List;

 import org.apache.http.HttpResponse;

 import org.apache.http.HttpStatus;

 import org.apache.http.NameValuePair;

 import org.apache.http.client.ClientProtocolException;

 import org.apache.http.client.HttpClient;

 import org.apache.http.client.entity.UrlEncodedFormEntity;

 import org.apache.http.client.methods.HttpPost;

 import org.apache.http.impl.client.DefaultHttpClient;

 import org.apache.http.message.BasicNameValuePair;

 import org.apache.http.util.EntityUtils;

 public class PostMail {

 public static void main(String[] args) throws IOException {

 final String url = "https://api.mailbrid.com/apiv1/send";

 final String accessKeyID = "您账户中的accessKeyID ";

 final String accessKey = "对应的accessKey ";

 final string loginname = "你注册时的邮箱地址";

 final string mobile = "你注册时的手机号码";

 final string timestr = DateTime.Now.ToString ("yyyy-MM-dd HH:mm:ss");

 final string accessToken = getMD5String(accessKeyID + accessKey + loginname + mobile + timestr);

 HttpClient httpclient = new DefaultHttpClient();

 HttpPost httPost = new HttpPost(url);

 List params = new ArrayList();

 params.add(new BasicNameValuePair("accessKeyID ", accessKeyID ));

 params.add(new BasicNameValuePair("accessToken ", accessToken ));

 params.add(new BasicNameValuePair("taskname", "任务名称"));

 params.add(new BasicNameValuePair("title", "邮件标题"));

 params.add(new BasicNameValuePair("summary", ""));

params.add(new BasicNameValuePair("content", "邮件内容"));

params.add(new BasicNameValuePair("tomail", "收件地址列表"));

params.add(new BasicNameValuePair("fromdns", "发件地址"));

params.add(new BasicNameValuePair("reply", ""));

params.add(new BasicNameValuePair("sendername", "发件人姓名"));

params.add(new BasicNameValuePair("pipeid", "通道id"));

params.add(new BasicNameValuePair("sendtime", timestr ));

 httpost.setEntity(new UrlEncodedFormEntity(params, "UTF-8"));

 // 请求

 HttpResponse response = httpclient.execute(httPost);

 // 处理响应

 if (response.getStatusLine().getStatusCode() == HttpStatus.SC\_OK) { // 正常返回

 // 读取xml文档

 String result = EntityUtils.toString(response.getEntity());

 System.out.println(result);

 } else {

 System.err.println("error");

 }

 httpost.releaseConnection();

 }

 public static String getMD5String(String str) {

 try {

 // 生成一个MD5加密计算摘要

 MessageDigest md = MessageDigest.getInstance("MD5");

 // 计算md5函数

 md.update(str.getBytes());

 // digest()最后确定返回md5 hash值，返回值为8位字符串。因为md5 hash值是16位的hex值，实际上就是8位的字符

 // BigInteger函数则将8位的字符串转换成16位hex值，用字符串来表示；得到字符串形式的hash值

 //一个byte是八位二进制，也就是2位十六进制字符（2的8次方等于16的2次方）

 return new BigInteger(1, md.digest()).toString(16);

 } catch (Exception e) {

 e.printStackTrace();

 return null;

 }

 }

 }

1. **示例代码:php**

 <?php

 function send\_mail() {

 $url = 'https://api.mailbrid.com/apiv1/send';

 $accessKeyID = '您账户中的accessKeyID';

 $accessKey = '所对应的accessKey';

 $ loginname = "你注册时的邮箱地址";

 $mobile = "你注册时的手机号码";

 $timestamp=time();

$timestr=date('Y-m-d H:i:s',$timestamp);

 $accessToken = md5(accessKeyID. accessKey . loginname . mobile . timestr);

 //您需要登录mailbrid创建accessKeyID，使用accessKeyID和accessKey才可以进行邮件的发送。

 $param = array(

 'accessKeyID' => $accessKeyID,

 'accessToken' => $accessToken,

 ‘taskname=> '任务名称',

 'title' => '邮件标题',

 'summary' => '邮件摘要',

 'content' => ‘邮件内容',

 'tomail' => ‘收件地址列表',

 'fromdns' => ‘发件地址',

'reply' => ‘回复地址',

'sendername' => ‘发件人名称',

 'pipeid' => ‘通道id',

 sendtime=> ‘发送时间'

 );

 $data = http\_build\_query($param);

 $options = array(

 'http' => array(

 'method' => 'POST',

 'header' => 'Content-Type: application/x-www-form-urlencoded',

 'content' => $data

 ));

 $context = stream\_context\_create($options);

 $result = file\_get\_contents($url, false, $context);

 return $result;

 }

 echo send\_mail();

 ?>

#### **获取发信通道接口**

##### 1)接口相关说明

接口地址:https://api.mailbrid.com/apiv1/getpipe

数据提交方式:get

参数说明

|  |  |  |
| --- | --- | --- |
| **参数名称** | **类型** | **备注** |
| accessKeyID | string | 平台中获取 |
| accessToken | string | 产生方法:accessToken=md5(accessKeyID+accessKey+注册时的email+注册时的手机号码+sendtime参数) |
| time | datetime | 时间字符串，产生accessToken用 |

#### **获取统计报表接口**

##### 1)接口相关说明

接口地址:https://api.mailbrid.com/apiv1/getreport

数据提交方式:get

参数说明

|  |  |  |
| --- | --- | --- |
| **参数名称** | **类型** | **备注** |
| accessKeyID | string | 平台中获取 |
| accessToken | string | 产生方法:accessToken=md5(accessKeyID+accessKey+注册时的email+注册时的手机号码+sendtime参数) |
| time | datetime | 时间字符串，产生accessToken用 |
| requestid | String  | 请求的发送任务id |

##### **返回数据格式说明**

返回json格式如下:

[

 {

 "flag": "string",

 "info": "string",

 "mailBody": {

 "taskname": "string",

 "title": "string",

 "summary": "string",

 "content": "string",

 "fromdns": "string",

 "reply": "string",

 "sendername": "string",

 "sendtime": "string",

 "status": "string"

 },

 "report": {

 "total": 0,

 "success": 0,

 "fail": 0,

 "nomailbox": 0,

 "opened": 0,

 "clicked": 0

 },

 "domainReport": [

 {

 "domain": "string",

 "total": 0,

 "success": 0,

 "open": 0,

 "click": 0

 }

 ],

 "areaReport": [

 {

 "country": "string",

 "opentimes": 0,

 "openpersons": 0,

 "clicktimes": 0,

 "clickpersons": 0

 }

 ]

 }

]

mailBody: 邮件信息

Report：邮件整体报表

domainReport：域名报表

areaReport：地域报表