邮件推送

SMTP接口说明

阿里云
aliyun.com

SMTP接口说明

SMTP服务地址

SMTP服务地址

SMTP服务地址: smtpdm.aliyun.com
SMTP端口号: 25，465（SSL加密）

SMTP之JAVA调用示例

使用javamail通过smtp协议发信

```
import javax.mail.*;
import javax.mail.internet.InternetAddress;
import javax.mail.internet.MimeMessage;
import java.util.Properties;

public class SimpleAliDMSendMail {
    private static final String ALIDM_SMTP_HOST = "smtpdm.aliyun.com";
    private static final int ALIDM_SMTP_PORT = 25;

    public static void main(String[] args) throws MessagingException {
        // 配置发送邮件的环境属性
        final Properties props = new Properties();
        // 表示SMTP发送邮件，需要进行身份验证
        props.put("mail.smtp.auth", "true");
        props.put("mail.smtp.host", ALIDM_SMTP_HOST);
        props.put("mail.smtp.port", ALIDM_SMTP_PORT);
        // 如果使用ssl，则去掉使用25端口的配置，进行如下配置,
        // props.put("mail.smtp.socketFactory.class", "javax.net.ssl.SSLSocketFactory");
        // props.put("mail.smtp.socketFactory.port", "465");
        // props.put("mail.smtp.port", "465");


        // 发件人的账号
        props.put("mail.user", "***");
        // 访问SMTP服务时需要提供的密码
        props.put("mail.password", "***");

        // 构建授权信息，用于进行SMTP进行身份验证
        Authenticator authenticator = new Authenticator() {
            @Override
            protected PasswordAuthentication getPasswordAuthentication() {
                // 用户名、密码
                String userName = props.getProperty("mail.user");
                String password = props.getProperty("mail.password");
                return new PasswordAuthentication(userName, password);
            }
        };
        // 使用环境属性和授权信息，创建邮件会话
        Session mailSession = Session.getInstance(props, authenticator);
        // 创建邮件消息
        MimeMessage message = new MimeMessage(mailSession);
        // 设置发件人
        InternetAddress form = new InternetAddress(
                props.getProperty("mail.user"));
        message.setFrom(form);

        // 设置收件人
        InternetAddress to = new InternetAddress("***");
        message.setRecipient(MimeMessage.RecipientType.TO, to);

        // 设置邮件标题
        message.setSubject("测试邮件");
        // 设置邮件的内容体
        message.setContent("测试的HTML邮件", "text/html;charset=UTF-8");

        // 发送邮件
        Transport.send(message);
    }
}
```

SMTP之CSharp调用示例

通过smtp协议发送带附件的邮件

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Net.Mail;
using System.Net.Mime;

namespace ConsoleApp
{
    class Program
    {
        static void Main(string[] args)
        {
            try
            {
                MailMessage mailMsg = new MailMessage();
                mailMsg.To.Add(new MailAddress("目标地址"));
                mailMsg.From = new MailAddress("控制台创建的发信地址", "昵称");
                // 邮件主题
                mailMsg.Subject = "邮件主题C#测试";
                // 邮件正文内容
                string text = "欢迎使用阿里云邮件推送";
                string html = @"欢迎使用<a href=""https://dm.console.aliyun.com"">邮件推送</a>";
                mailMsg.AlternateViews.Add(AlternateView.CreateAlternateViewFromString(text, null, MediaTypeNames.Text.Plain));
                mailMsg.AlternateViews.Add(AlternateView.CreateAlternateViewFromString(html, null, MediaTypeNames.Text.Html));
                // 添加附件
                string file = "D:\\1.txt";
                Attachment data = new Attachment(file, MediaTypeNames.Application.Octet);
                mailMsg.Attachments.Add(data);
                //邮件推送的SMTP地址和端口
                SmtpClient smtpClient = new SmtpClient("smtpdm.aliyun.com", 25);
                // 使用SMTP用户名和密码进行验证
                System.Net.NetworkCredential credentials = new System.Net.NetworkCredential("控制台创建的发信地址
", "控制台设置的SMTP密码");
                smtpClient.Credentials = credentials;
                smtpClient.Send(mailMsg);
            }
            catch (Exception ex)
            {
                Console.WriteLine(ex.ToString());
            }
        }
    }

}
```

SMTP之PHP调用示例

使用php通过smtp协议发信

```
<?php
require 'email.class.php';

$mailto='***';
$mailsubject="测试邮件";
$mailbody='这里是邮件内容';

$smtpserver     = "smtpdm.aliyun.com";
$smtpserverport = 25;
$smtpusermail   = "***";
$smtpuser       = "***";
$smtppass       = "***";

$mailsubject    = "=?UTF-8?B?" . base64_encode($mailsubject) . "?=";
$mailtype       = "HTML";
$smtp           = new smtp($smtpserver, $smtpserverport, true, $smtpuser, $smtppass);
$smtp->debug    = false;
$smtp->sendmail($mailto, $smtpusermail, $mailsubject, $mailbody, $mailtype);
```

注：email.class.php 为外部依赖，请自行获取。 smtpuser = “请替换完整的邮件推送地址”

SMTP之python调用示例

使用python通过smtp协议发信

```
### -*- coding: utf-8 -*-
```

```
from email.header import Header
from email.mime.text import MIMEText
import smtplib

from_addr = "***"
password = "***"
to_addr = "***"
smtp_server = "smtpdm.aliyun.com"

msg = MIMEText('测试邮件内容', 'plain', 'utf-8')
msg['From'] = from_addr
msg['To'] = to_addr
msg['Subject'] = Header(u'测试邮件', 'utf-8').encode()

server = smtplib.SMTP(smtp_server, 25)
server.set_debuglevel(1)
server.login(from_addr, password)
server.sendmail(from_addr, [to_addr], msg.as_string())
server.quit()
```

SMTP之perl调用示例

使用perl通过smtp协议发信

```
### Swaks is an smtp of CURL, install it first:
curl http://www.jetmore.org/john/code/swaks/files/swaks-20130209.0/swaks -o swaks
### Set the permissions for the script so you can run it
chmod +x swaks
### It's based on perl, so install perl
sudo apt-get -y install perl
### now send!
./swaks --auth \
        --server smtpdm.aliyun.com \
        --au domaintest@dm.aliyun.com \
        --ap ***** \
        --from domaintest@dm.aliyun.com \
        --to test@test.com \
        --h-Subject: "Hello" \
        --body 'Testing mail!'
```

注：请在 --au 后替换您的邮件推送地址；--ap 后替换您的邮件推送密码；--from后替换邮件推送地址；--to 后替换接收地址；

SMTP之Ruby调用示例

使用Ruby通过smtp协议发信

```
### install `mail` gem first: `gem install mail`

require 'mail'
```

```
Mail.defaults do
  delivery_method :smtp, {
    :port      => 25,
    :address   => "smtpdm.aliyun.com",
    :user_name => "domaintest@dm.aliyun.com",
    :password  => "****",
    :enable_starttls_auto => false,
    :openssl_verify_mode => 'none',
  }
end

mail = Mail.deliver do
  to      'test@test.com'
  from    'domaintest@dm.aliyun.com'
  subject 'Hello'

  text_part do
    body 'Testing mail'
  end
end
```

注：请在 :user_name => 后替换您的邮件推送地址；:password => 后替换您的邮件推送密码；to '后替换接收地址；from '后替换邮件推送地址

SMTP之nodejs调用示例

使用nodejs通过smtp协议发信

```
// load nodemailer as follows
// npm install nodemailer --save

var nodemailer = require('nodemailer');

// create reusable transporter object using SMTP transport
var transporter = nodemailer.createTransport({
    "host": "smtpdm.aliyun.com",
    "port": 25,
    "secureConnection": true, // use SSL
    "auth": {
        "user": 'username@userdomain', // user name
        "pass": 'xxxxxxx'         // password
    }
});

// NB! No need to recreate the transporter object. You can use
// the same transporter object for all e-mails

// setup e-mail data with unicode symbols
var mailOptions = {
    from: 'NickName<username@userdomain>', // sender address mailfrom must be same with the user
    to: 'x@x.com, xx@xx.com', // list of receivers
    cc:'haha<xxx@xxx.com>', // copy for receivers
    bcc:'haha<xxxx@xxxx.com>', // secret copy for receivers
    subject: 'Hello', // Subject line
    text: 'Hello world', // plaintext body
    html: '<b>Hello world</b><img src="cid:01" style="width:200px;height:auto">', // html body
    attachments: [
        {
            filename: 'text0.txt',
            content: 'hello world!'
        },
        {
            filename: 'text1.txt',
            path: './app.js'
        },{
            filename:'test.JPG',
            path:'./Desert.jpg',
            cid:'01'
      }
    ],

};

// send mail with defined transport object
transporter.sendMail(mailOptions, function(error, info){
    if(error){
        return console.log(error);
    }
    console.log('Message sent: ' + info.response);

});
```

SMTP之GO调用示例

使用GO语言通过smtp协议发送邮件

```
package main

import (
    "fmt"
    "net/smtp"
    "strings"
)

func SendToMail(user, password, host, to, subject, body, mailtype string) error {
    hp := strings.Split(host, ":")
    auth := smtp.PlainAuth("", user, password, hp[0])
    var content_type string
    if mailtype == "html" {
        content_type = "Content-Type: text/" + mailtype + "; charset=UTF-8"
    } else {
        content_type = "Content-Type: text/plain" + "; charset=UTF-8"
    }

    msg := []byte("To: " + to + "\r\nFrom: " + user + "\r\nSubject: " + subject + "\r\n" + content_type + "\r\n\r\n" +
body)
    send_to := strings.Split(to, ";")
    err := smtp.SendMail(host, auth, user, send_to, msg)
    return err
}

func main() {
    user := "控制台创建的发信地址"
    password := "控制台设置的SMTP密码"
    host := "smtpdm.aliyun.com:25"
    to := "目标地址"

    subject := "test Golang to sendmail"

    body := `
        <html>
        <body>
        <h3>
        "Test send to email"
        </h3>
        </body>
        </html>
        `
    fmt.Println("send email")
    err := SendToMail(user, password, host, to, subject, body, "html")
    if err != nil {
        fmt.Println("Send mail error!")
        fmt.Println(err)
    } else {
        fmt.Println("Send mail success!")
    }

}
```