# 尚马教育 JAVA 高级课程

# 短信与邮件

文档编号：C12

创建日期： 2017-07-07

最后修改日期：2021-02-26

版 本 号：V3.5

电子版文件名：尚马教育-第三阶段-12.短信与邮件.docx

**文档修改记录：**

|  |  |  |  |
| --- | --- | --- | --- |
| 更新日期 | 更新作者 | 更新说明 | 版本号 |
| 2017-07-30 | 张元林 | 初始版本 | V1.0 |
| 2018-08-01 | 王绍成 | Mybatis版本更新 | V2.0 |
| 2019-08-09 | 冯勇涛 | 课件格式以及课程深度加深 | V3.0 |
| 2021-01-26 | 冯勇涛 | 修改阿里云短信 | V3.5 |

目录

[1. Sms](#_Toc12067_WPSOffice_Level1) [3](#_Toc12067_WPSOffice_Level1)

[1.1. 使用场景](#_Toc25785_WPSOffice_Level2) [3](#_Toc25785_WPSOffice_Level2)

[1.2. 短信猫](#_Toc30359_WPSOffice_Level2) [3](#_Toc30359_WPSOffice_Level2)

[1.3. 云短信服务](#_Toc22305_WPSOffice_Level2) [3](#_Toc22305_WPSOffice_Level2)

[1.3.1. 短信API](#_Toc25785_WPSOffice_Level3) [3](#_Toc25785_WPSOffice_Level3)

[1.3.2. 查询发送信息](#_Toc30359_WPSOffice_Level3) [3](#_Toc30359_WPSOffice_Level3)

[2. 邮件发送](#_Toc25785_WPSOffice_Level1) [3](#_Toc25785_WPSOffice_Level1)

[发送协议：SMTP](#_Toc30359_WPSOffice_Level1) [3](#_Toc30359_WPSOffice_Level1)

[接收协议：POP3](#_Toc22305_WPSOffice_Level1) [3](#_Toc22305_WPSOffice_Level1)

## Sms

### 使用场景

Java发送短信的功能是通过其他设备来实现的，同样是需要付费。

在早期，比较流行使用**短信猫**来发送短信，短信猫相当于一个简单的手机，可以插入SIM卡，通过Java串口编程来调用短信猫的AT指令，再由短信猫来发送短信。但是受限于设备的性能，需要大量发送短信的时候，要购买更好的设备来支持，成本较高。

近年来互联网上出现了许多短信发送平台，速度更快，收费更低，成本上远远优于自己搭建的短信平台，而且更加稳定。

市场上短信发送的平台有很多，每家公司会根据自身的资源与不同的短信平台合作，但是从程序角度出发，所有短信平台的使用流程大体相同。

当前文档以阿里云的短信平台为例，介绍一下如何使用Java发送短信。

### 短信猫

短信猫，其实是一种用来收发短信的设备，他和我们用的手机一样，需要手机[SIM卡](https://baike.baidu.com/item/SIM%E5%8D%A1" \t "https://baike.baidu.com/item/%E7%9F%AD%E4%BF%A1%E7%8C%AB/_blank)的支持，在需要收发短信的时候，在短信猫里面插入一张我们平时用的手机卡，插上电源，通过（USB或者串口、网口）数据线和电脑相连，在电脑的应用[管理软件](https://baike.baidu.com/item/%E7%AE%A1%E7%90%86%E8%BD%AF%E4%BB%B6" \t "https://baike.baidu.com/item/%E7%9F%AD%E4%BF%A1%E7%8C%AB/_blank)中就可以实现短信收发的功能。



### 云短信服务

这一步一般应该由运维人员搞定，但是多知道一些是没有坏处的。

阿里云的官网地址:<https://www.aliyun.com>

如果是为公司申请，最好让其他人员申请一个公司账户，或者老板的账户，建议公司账户升级为企业用户，而不是使用个人账户，不然离职之后，很麻烦。

个人学习，可以直接使用淘宝登录。

|  |
| --- |
|  |

登录成功后，点击首页右上角的控制台

|  |
| --- |
|  |

搜索短信服务

|  |
| --- |
|  |
|  |

使用短信平台之前应该先充值，保证账户里的余额可以发送短信。

|  |
| --- |
|  |

快速学习

|  |
| --- |
|  |

#### 短信API

##### 添加签名与模板

|  |
| --- |
|  |
| 签名和模板都搞定之后，就可以进一步研究如何使用Java程序调用 |

##### 查看api

|  |
| --- |
|  |

##### 获取授权码

什么是AccessKey？网站中有详细的介绍，简单点说，就是程序调用的一个用户名密码，所以必须保密，其他人知道这个AccessKey之后，就可以调用你的个人用户了，使用你的模板发送短信。

|  |
| --- |
|  |
| 记住这个AccessKeyID和AccessKeySecret 一会写代码的时候会用到 |

##### 添加maven依赖

|  |
| --- |
| <dependency>  <groupId>com.aliyun</groupId>  <artifactId>aliyun-java-sdk-core</artifactId>  <version>4.5.3</version>  </dependency> |

##### 代码示例

|  |
| --- |
| **public** **static** **void** send(String phone,String code) {  DefaultProfile profile = DefaultProfile.*getProfile*("cn-hangzhou", "待填写", "待填写");  IAcsClient client = **new** DefaultAcsClient(profile);  CommonRequest request = **new** CommonRequest();  request.setMethod(MethodType.***POST***);  request.setDomain("dysmsapi.aliyuncs.com");  request.setVersion("2017-05-25");  request.setAction("SendSms");  request.putQueryParameter("RegionId", "cn-hangzhou");  request.putQueryParameter("PhoneNumbers",phone);  request.putQueryParameter("SignName", "禾十六美肌整骨社");  request.putQueryParameter("TemplateCode", "SMS\_172738690");  request.putQueryParameter("TemplateParam", "{\"code\":"+code+"}");  **try** {  CommonResponse response = client.getCommonResponse(request);  System.***out***.println(response.getData());  } **catch** (ServerException e) {  e.printStackTrace();  } **catch** (ClientException e) {  e.printStackTrace();  }  } |

#### 查询发送信息

|  |
| --- |
| **public** **static** **void** main(String[] args) {  DefaultProfile profile = DefaultProfile.*getProfile*("cn-hangzhou", "待填写", "待填写");  IAcsClient client = **new** DefaultAcsClient(profile);  CommonRequest request = **new** CommonRequest();  request.setMethod(MethodType.***POST***);  request.setDomain("dysmsapi.aliyuncs.com");  request.setVersion("2017-05-25");  request.setAction("QuerySendDetails");  request.putQueryParameter("RegionId", "cn-hangzhou");  request.putQueryParameter("PhoneNumber", "13767599447");  request.putQueryParameter("SendDate", "20191010");  request.putQueryParameter("PageSize", "10");  request.putQueryParameter("CurrentPage", "1");  **try** {  CommonResponse response = client.getCommonResponse(request);  System.***out***.println(response.getData());  } **catch** (ServerException e) {  e.printStackTrace();  } **catch** (ClientException e) {  e.printStackTrace();  }  } |

## 邮件发送

发送协议：SMTP

接收协议：POP3

### 代码示例

|  |
| --- |
| String account="XXXXX@163.com";  String auth="abc123456789";    //SMTP协议内容。  Properties properties = **new** Properties();  properties.put("mail.transport.protocol", "smtp");// 连接协议  properties.put("mail.smtp.host", "smtp.163.com");// 主机名 smtp.163.com  properties.put("mail.smtp.port", 465);// 默认端口号 25  properties.put("mail.smtp.auth", "true");//服务端认证。  properties.put("mail.smtp.ssl.enable", "true");// 设置是否使用ssl安全连接 ---一般都使用  properties.put("mail.debug", "true");// 设置是否显示debug信息 true 会在控制台显示相关信息  Session session = Session.*getDefaultInstance*(properties);//与邮件服务器的会话对象  **try** {  Address[] to = **new** Address[]{**new** InternetAddress("XXXX@qq.com")};//接收人  Message msg = **new** MimeMessage(session);//主体，接收人，内容，附件。。。  msg.setFrom(**new** InternetAddress(account));//发送人  msg.setRecipients(RecipientType.***TO***, to);//接收人()  msg.setSubject("测试邮件");//标题  msg.setText("邮件内容");//内容    Transport transport = session.getTransport();//邮差对象  transport.connect(account, auth);    transport.sendMessage(msg, msg.getAllRecipients());  } **catch** (Exception e) {  e.printStackTrace();  } |

### 封装工具类

详见:邮件工具类 文件夹

MailSender：

|  |
| --- |
| package com.javasm.util.email;  import java.io.File;  import java.util.ArrayList;  import java.util.Date;  import java.util.List;  import java.util.Properties;  import javax.activation.DataHandler;  import javax.activation.DataSource;  import javax.activation.FileDataSource;  import javax.mail.Message;  import javax.mail.Session;  import javax.mail.Transport;  import javax.mail.internet.InternetAddress;  import javax.mail.internet.MimeBodyPart;  import javax.mail.internet.MimeMessage;  import javax.mail.internet.MimeMultipart;  import javax.mail.internet.MimeUtility;  public class MailSender  {  /\*\* 默认 Content Type -> text/plain \*/  public static final String DEFAULT\_CONTENT\_TYPE = "text/plain";  /\*\* 默认字符编码 -> UTF-8 \*/  public static final String DEFAULT\_ENCODING = GeneralHelper.DEFAULT\_ENCODING;  /\*\* 默认 SMTP 端口 -> 25 \*/  public static final int DEFAULT\_PORT = 25;    private String host = "";  private String from = "";  private String user = "";  private String password = "";  private String subject = "";  private String text = "";  private String contentType = DEFAULT\_CONTENT\_TYPE;  private String charset = DEFAULT\_ENCODING;  private int port = DEFAULT\_PORT;  private boolean auth = true;  private boolean needReceipt = false;  private Date sentDate = null;    private List<String> to = new ArrayList<String>();  private List<String> cc = new ArrayList<String>();  private List<String> bcc = new ArrayList<String>();  private List<String> replyTo = new ArrayList<String>();  private List<String> fileAcc = new ArrayList<String>();  private List<MimeBodyPart> byteAcc = new ArrayList<MimeBodyPart>();    public int getPort()  {  return port;  }  public void setPort(int port)  {  this.port = port;  }    public boolean isAuth()  {  return auth;  }  public void setAuth(boolean auth)  {  this.auth = auth;  }  public String getCharset()  {  return charset;  }  public void setCharset(String charset)  {  this.charset = charset;  }  public String getContentType()  {  return contentType;  }  public void setContentType(String contentType)  {  this.contentType = contentType;  }  public boolean isNeedReceipt()  {  return needReceipt;  }  public void setNeedReceipt(boolean needReceipt)  {  this.needReceipt = needReceipt;  }  public String getFrom()  {  return from;  }  public void setFrom(String from)  {  this.from = from;  }  public String getHost()  {  return host;  }  public void setHost(String host)  {  this.host = host;  }  public String getPassword()  {  return password;  }  public void setPassword(String password)  {  this.password = password;  }  public String getSubject()  {  return subject;  }  public void setSubject(String subject)  {  this.subject = subject;  }  public String getText()  {  return text;  }  public void setText(String text)  {  this.text = text;  }    public Date getSentDate()  {  return sentDate;  }    public void setSentDate(Date sentDate)  {  this.sentDate = sentDate;  }  public String getUser()  {  return user;  }  public void setUser(String user)  {  this.user = user;  }  public List<String> getFileAcc()  {  return fileAcc;  }    public List<MimeBodyPart> getByteAcc()  {  return byteAcc;  }    public void setFileAcc(List<String> accessory)  {  this.fileAcc = accessory;  }    public void setByteAcc(List<MimeBodyPart> accessory)  {  this.byteAcc = accessory;  }  public List<String> getReplyTo()  {  return replyTo;  }  public List<String> getTo()  {  return to;  }    public void setTo(List<String> to)  {  this.to = to;  }    public List<String> getCc()  {  return cc;  }    public void setCc(List<String> cc)  {  this.cc = cc;  }    public List<String> getBcc()  {  return bcc;  }    public void setBcc(List<String> bcc)  {  this.bcc = bcc;  }    public void addFileAcc(String accessory)  {  fileAcc.add(accessory);  }    /\*\* 添加 byte array 形式的附件 \*/  public void addByteAcc(byte[] accessory, String type, String fileName) throws Exception  {  ByteArrayDataSource ds = new ByteArrayDataSource(accessory, type, fileName);  //BASE64Encoder enc = new BASE64Encoder();  //fileName = "=?GBK?B?" + enc.encode(fileName.getBytes()) + "?=";  fileName = MimeUtility.encodeText(fileName, charset, "B");  MimeBodyPart mimeFile = new MimeBodyPart();  mimeFile.setDataHandler(new DataHandler(ds));  //mimeFile.setFileName(ds.getName());  mimeFile.setFileName(fileName);    byteAcc.add(mimeFile);  }    public void addReplyTo(String address)  {  replyTo.add(address);  }    public void addTo(String address)  {  to.add(address);  }    public void addCc(String address)  {  cc.add(address);  }    public void addBcc(String address)  {  bcc.add(address);  }    /\*\* 发送邮件 \*/  public void send() throws Exception  {  Transport transport = null;    try  {  Properties props = new Properties();  props.put("mail.transport.protocol", "smtp");  props.put("mail.smtp.auth", Boolean.toString(auth));  Session session = Session.getDefaultInstance(props, null);  MimeMessage msg = new MimeMessage(session);  msg.setFrom(new InternetAddress(from));    for(String i : to)  msg.addRecipient(Message.RecipientType.TO, new InternetAddress(i));  for(String i : cc)  msg.addRecipient(Message.RecipientType.CC, new InternetAddress(i));  for(String i : bcc)  msg.addRecipient(Message.RecipientType.BCC, new InternetAddress(i));  if(replyTo.size() > 0)  {  InternetAddress[] replyAddress = new InternetAddress[replyTo.size()];    for(int i = 0; i < replyAddress.length; i++)  replyAddress[i] = new InternetAddress((String)replyTo.get(i));    msg.setReplyTo(replyAddress);  }    if(needReceipt)  msg.addHeader("Disposition-Notification-To", from);    if(sentDate != null)  msg.setSentDate(sentDate);  else  msg.setSentDate(new Date());    msg.setSubject(subject, charset);    MimeMultipart mm = new MimeMultipart();  MimeBodyPart mbText = new MimeBodyPart();  mbText.setContent(text, contentType + ";charset=" + charset);  mm.addBodyPart(mbText);    for(String filePath : fileAcc)  {  String fileName = (new File(filePath)).getName();  fileName = MimeUtility.encodeText(fileName, charset, "B");    MimeBodyPart mbFile = new MimeBodyPart();  DataSource datasource = new FileDataSource(filePath);    mbFile.setDataHandler(new DataHandler(datasource));  mbFile.setFileName(fileName);  mm.addBodyPart(mbFile);  }    for(MimeBodyPart part : byteAcc)  mm.addBodyPart(part);    msg.setContent(mm);  msg.saveChanges();    transport = session.getTransport();  transport.connect(host, port, user, password);  transport.sendMessage(msg, msg.getAllRecipients());  }  finally  {  if(transport != null) try{ transport.close(); } catch (Exception e) { }  }  }  } |