

## 一、引言

**实验时间：**2019 年 10 月 14（周一）18:30-21:30，共 4 课时

**实验目的：**掌握开发、测试、发布、调用 Web Service 的基本方法、工具和流程，理解 Web Service 风格软件架构基本原理、结构和特点。

**背景及要求：**

*[综述研究背景：概述本项工作的研究或观察的理论基础，给出简明的理论或研究背景，一定要列举重要的相关文献。若可能指出存在问题：说明为什么要做这项工作；阐述研究目的：说明有别于他人的“主意”（此红色字体一条不做强行要求）。]*

现实生活中，存在着大量发送手机短信通知，发送电子邮件消息的系统需求，例如：当你登录 AWS 或阿里云账号时，可以通过手机验证实现账号登入功能。这类通知服务是可以在其他应用程序中重用的功能模块。

1. 结合课堂上讲授的 SOA 风格，基于 AWS SES 或阿里云的邮件服务，实现一个能发送电子邮件消息的 Web Service 服务，包括如下三个具体服务（开发语言不限，要求提供基于 SOAP 协议和 REST 风格的两种接口）：

- `sendEmail(String _url,String _payload)` //邮件地址为\_url，内容为\_payload
- `sendEmailBatch(String[] _url,String _payload)` //批量发送邮件
- `validateEmailAddress(String _url)` //验证是否为有效的邮件地址

**注：**

a. 以上三个服务，返回的值为 Y 或者 N。例如在第一个服务中，发送成功则返回 Y，发送失败则返回 N。

b. 在你实现这个服务时，可基于 AWS SES 或阿里云的邮件推送服务来实现你的 Web Service，AWS SES 参考链接 <https://aws.amazon.com/cn/ses/>，阿里云的邮件推送服务地址 <https://dm.console.aliyun.com/>。

c. `validateEmailAddress` 方法尽量使用正则表达式完成。

2. 编写至少两种版本客户端（例如：桌面版、Web 版、Android 版、IOS 版），分别调用你编写的邮件推送 Web Service 服务，并分析 Web Service 集中解决远程调用、跨平台调用、跨语言调用所带来的好处及不足。

3. 结合上述实例，从软件体系结构风格的角度分析 Web Service 风格的主要构件和连接件？

提示：可参考课堂上讲授的“两层 C/S 结构”风格

**基本构件：**

- 数据库服务器：存放数据的数据库、负责数据处理的业务逻辑；
- 客户机应用程序：
- GUI：用户界面
- 业务逻辑：利用客户机上的应用程序对数据进行处理；

**连接件：**经由网络的调用-返回机制或事件机制。

- 客户机<->服务器：客户机向服务器发送请求，并接收返回结果。

4. 参考 Kruchten 4+1 视图模型，试着给出上述系统的视图模型。

## 二、实验设计（给出你的实习内容的设计方案，可根据实际情况调整条目）

## 2.1 系统需求

[介绍目标系统的技术环境需求、功能需求、质量需求（如果需要强调的话就给出）、约束（如果有的话就给出）等]

### 2.1.1 技术环境需求：

硬件环境：

操作系统：ubuntu16

Cpu：1G

软件环境：Java 环境

### 2.1.2 功能需求：

- 1) 实现三个具体服务，并且基于 SOAP 协议和 rest 风格都实现一次
- 2) 编写至少两种版本客户端

### 2.1.3 质量需求：

- 1) 实现基本的提供服务的功能
- 2) 客户端不会无缘无故崩溃

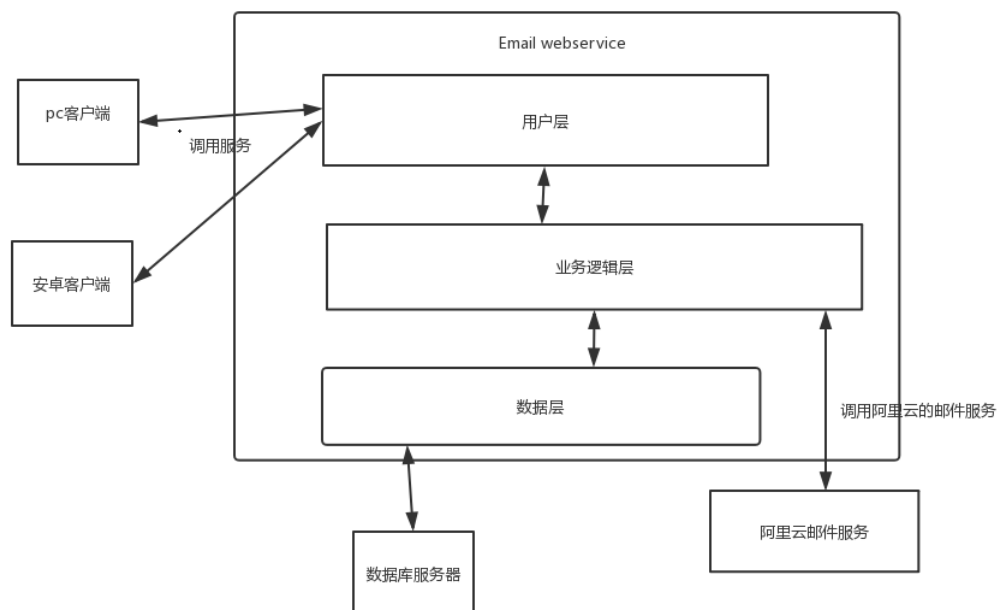
### 2.1.4 约束：

实现的 webservice 遵循 http 协议

## 2.2 架构设计

[给出软件架构图，并给出主要构件和连接件的文字说明]

Webservice 架构图（使用线框图表示）：



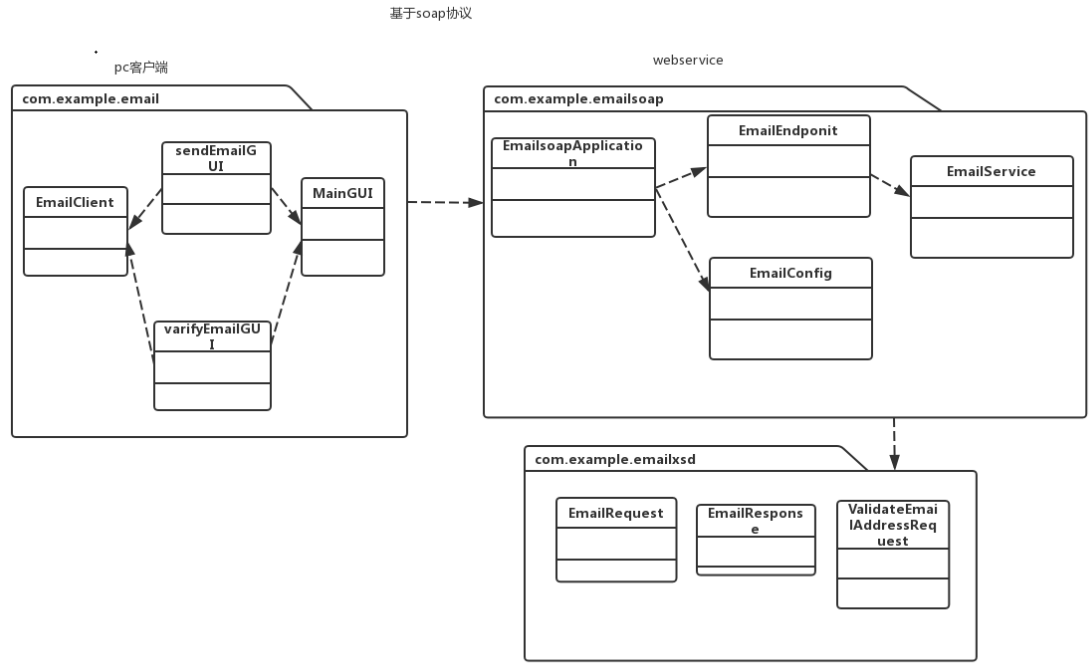
构件说明：

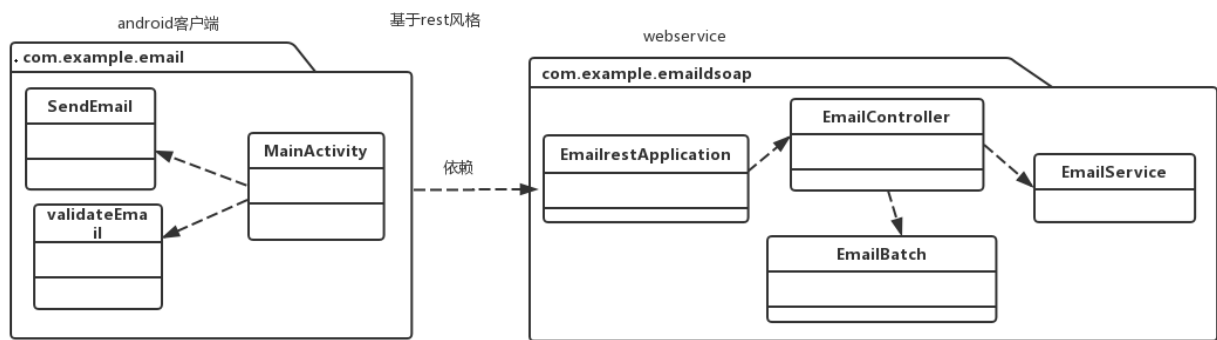
构件名	描述
Webservice 服务器	提供 webservice 服务的服务器，提供相对应的接口，给其他应用服务，此题
数据库服务器	存放数据的数据库、负责数据处理的业务逻辑
阿里云邮件服务	由阿里云提供的邮件服务，每天两百份免费，在 webservice 中调用，用于发送邮件
Pc 客户端	使用 java 写的客户端，在其中调用了写的 webservice 服务，用于给别人发送邮件、群发邮件以及验证邮件
Android 客户端	使用 java 写的 android 客户端，其中调用了 webservice 服务，用于给别人发送邮件、群发邮件。

连接件说明：

连接件名	描述
http 协议	Webservice 是建立在 http 协议之上，通过 http 连接了两个进程
SOAP 协议	在基于 SOAP 协议的 webservice，是通过 soap 协议通讯，SOAP 连接了两个进程

逻辑视图（包图）：





因为设计了两种所以在实现客户端的时候，采取 pc 端的使用基于 soap 协议的 webservice,而 android 端采用 rest 风格的 webservice，下面对每个包里面的类进行说明。

#### Pc 客户端：

类名	描述
MainGUI	主显示界面，进入程序之后显示的界面，可以在里面选择发送邮件还是验证邮箱
sendMailGUI	发送右键的界面，可以群发也可以单发
varifyMailGUI	验证邮箱的界面
EmailClient	真正的客户端类，在里面调用 webservice 进行发送邮件、验证邮箱等工作，三个界面调用里面的功能，所以三个界面都依赖这个类

基于 soap 协议的 webservice，采用的是 springboot 框架进行编写的，下面是对其的说明：

类名	描述
EmailsoapApplication	程序的启动类
EmailConfig	配置类，在里面对整个程序利用注解进行了配置
EmailEndponit	发布服务的类，在里面是对服务的发布
EmailService	服务类，在里面写了利用阿里云发布邮件的服务以及验证邮件的服务

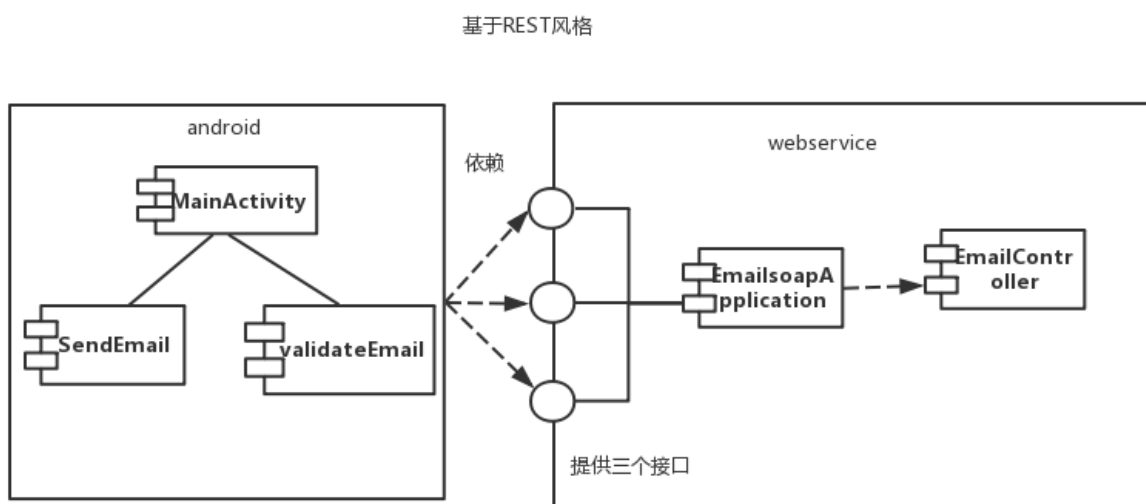
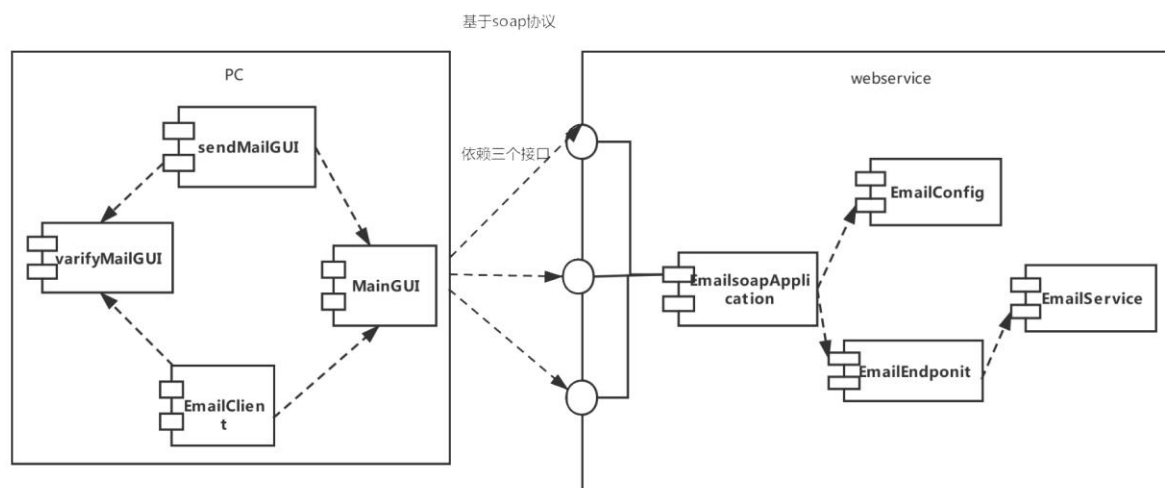
#### Android 客户端：

类名	描述
MainActivity	进入程序后的活动，在这个活动当中选择发送邮件还是验证邮箱
SendEmail	发送邮件活动，在里面可以选择群发或者单发
validateEmail	验证邮件服务，在里面可以进行邮箱的验证

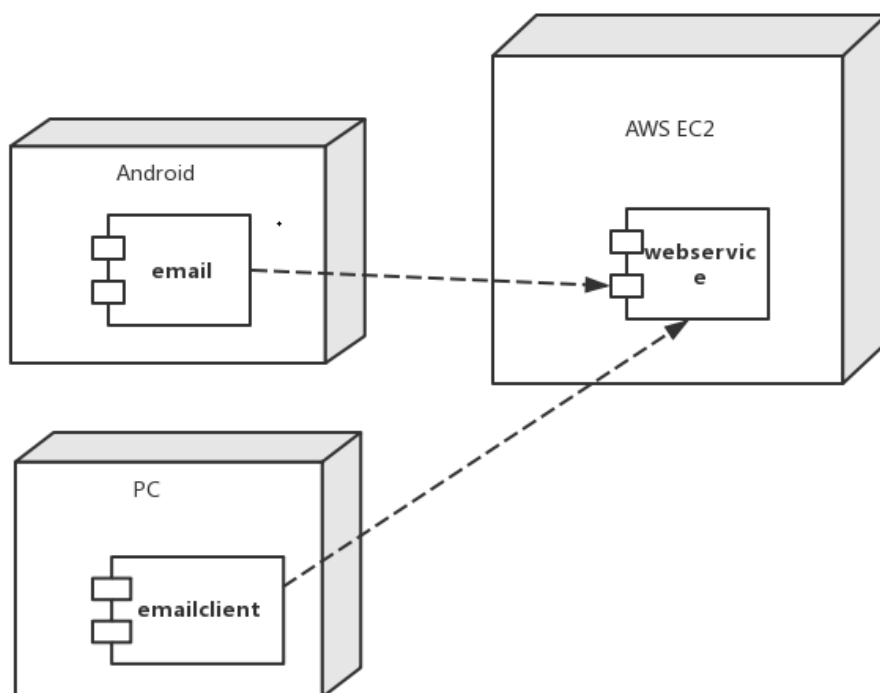
基于 rest 风格的 webservice,也使用了 springboot 框架进行边，下面是对其的说明：

类名	描述
EmailrestApplication	程序的启动类
EmailController	程序的控制类，并且利用注解在里面发布了服务
EmailService	服务类，在里面写了利用阿里云发布邮件的服务以及验证邮件的服务
EmailBatch	类似一个 Bean,用于接收群发时候的数组

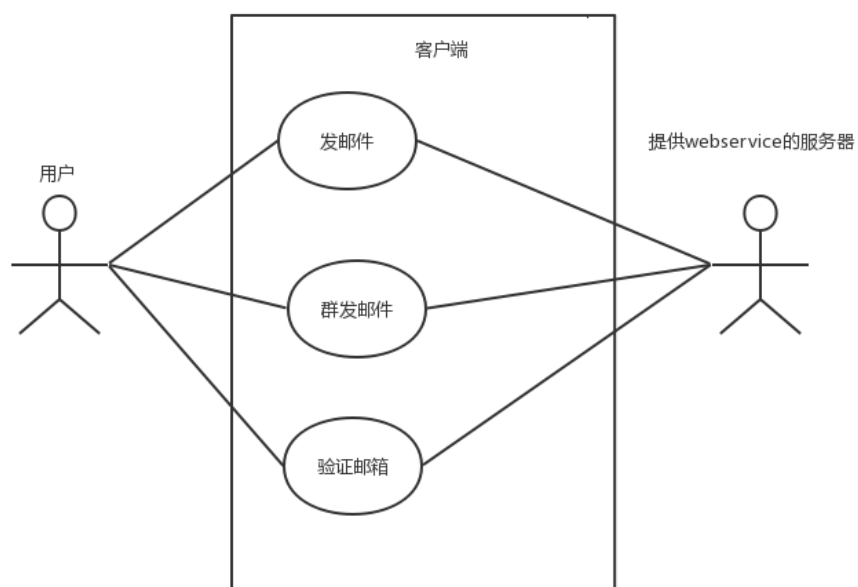
开发视图（构件图）：



物理视图（部署图）：



用例视图（用例图）：



## 2.3 接口设计

[给出系统主要服务接口设计]

基于 Soap 协议写的 webseervice，主要服务接口在 EmailEndponit 中

接口名	参数	返回值	说明
EmailResponse sendEmail(@RequestPayload EmailRequest request)	接收 EmailRequest 请 求，进行发送邮件	EmailResponse ，在其中有个 result 参数，说明发送成 功还是失败	该接口用 于发送邮 件，群发 邮箱接口
EmailResponse validateEmailAddress(@RequestPayload ValidateEmailAddressRequest request)	接收 ValidateEmail AddressRequest 请 求，进行发送邮件	EmailResponse ，在其中有个 result 参数，说明发送成 功还是失败	客户端调 用此接口 进行邮件 格式验证

基于 rest 风格写的 webseervice，主要接口在 EmailController 当中

接口名	参数	返回值	说明
public String sendEmail(@RequestParam(value = "_url",defaultValue = "1767508581@qq.com") String _url,  @RequestParam(value = "_payload",defaultValue = "hellow world") String _payload )	都是 String 类型 类型，_url 代表需 要发送邮件的地 址，_payload 代表 需要发送邮件的 内容	String 类型，如 果发送成功就 返回“Y”，如 果发送失败就 返回“N”。	客户端调用此 接口进行邮件 的单独发送
public String sendEmailBatch2(@RequestBody EmailBatch email)	EmailBatch 类型 (由 json 转化而 成)，该类型当中 有一个 string 数组 类型的地址数组 和 string 类型的内 容	String 类型，如 果发送成功就 返回“Y”，如 果发送失败就 返回“N”。	客户端调用此 接口进行邮件 的批量发送
public String validateEmailAddress(@RequestParam(value = "_url",defaultValue = "176") String _url)	接收一个 Stringl 类 型的_url，判断— _url 是否符合邮件 格式	String 类型，如 果发送成功就 返回“Y”，如 果发送失败就 返回“N”。	客户端调用此 接口进行邮箱 格式验证。

## 三、实验过程

### 3.1 软件实现

[说明系统开发、测试等与实现相关的技术]

系统开发：

Java 语言

实现相关技术：

Springboot

### 3.2 实验环境

[包括使用的硬件、软件（服务器/客户端操作系统，服务器、数据库、虚拟机等支撑软件）、实验场景（若有必要的话简单描述一下实验场景情况并给出照片）]

3.2.1 硬件：

Lenovo: R720

Cpu: 2.5GHZ,8 核

内存: 8G

硬盘: 1T(机械)+128G（固态）

3.2.2 软件：

编译器: IntelliJ IDEA 2019.2.2、Android Studio

### 3.3 实验步骤

[给出实验步骤，及每一步详细参数设置]

对于基于 soap 协议的 webservice

- ①在 idea 上建立一个 springboot 的项目，记得选中基于 soap 的 webservice
- ②写一个 Email.xsd，余姚符合格式要求，再通过工具将 xsd 反向生成 java 代码
- ③编写 EmailService 接口以及 EmailServiceImpl 实现类，在 EmailServiceImpl 中调用了阿里云的邮件服务
- ④编写配置文件类 EmailConfig，在其中对整个项目进行配置
- ⑤编写 EmailEndponit，在其中对于服务进行发送

对于基于 rest 风格的 webservice

- ①编写 EmailService 接口以及 EmailServiceImpl 实现类，在 EmailServiceImpl 中调用了阿里云的邮件服务
- ②编写 EmailBatch 类
- ③EmailController 在里面对于项目的服务进行配置以及发布



Pc 端

- ①编写 EmailClient 里面实现了使用 webservice 接口进行邮件发送、邮箱验证的基本功能
- ②编写 MainGUI、sendMailGUI、varifyMailGUI 三个界面类，再其中调用 EmailClient。

Android 端

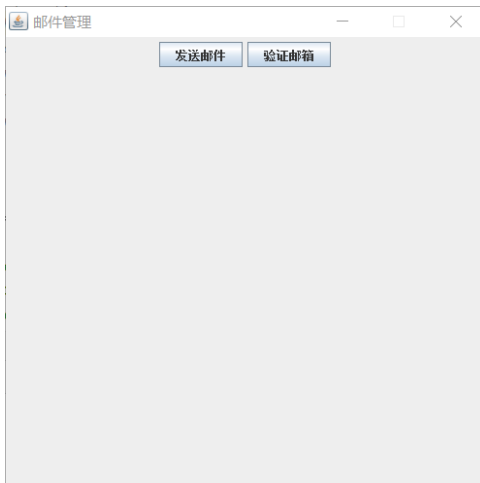
- ①编写 MainActivity 实现界面跳转
- ②编写 SendEmail，在里面调用 webservice 的接口实现发送邮件
- ③编写 validateEmail，在里面调用 webservice 的接口实现邮箱验证

## 四、实验评价

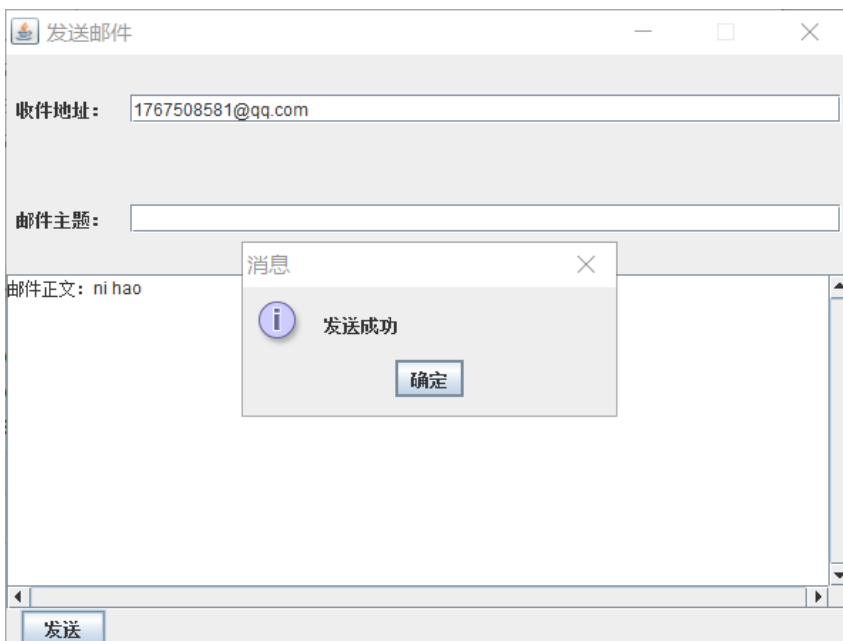
### 4.1 实验结果

[给出本次实验的结果和发现（必要时应使用图表，含软件运行结果截图）。]

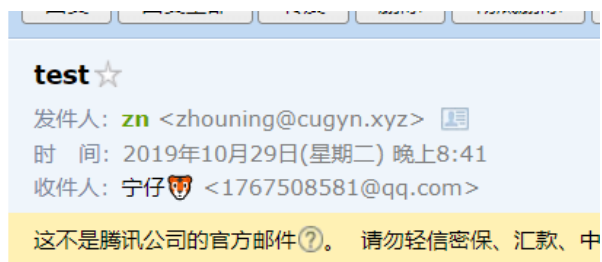
pc 客户端起始界面：



单发邮件给自己的 QQ 邮箱

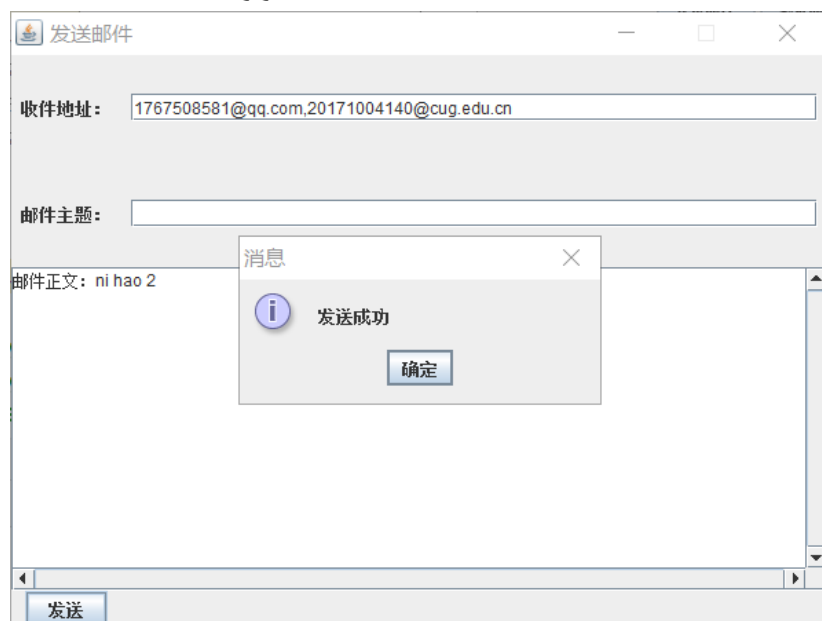


## QQ 邮箱收到的消息



邮件正文: ni hao

批量发送给自己的 QQ 邮箱和地大邮箱:

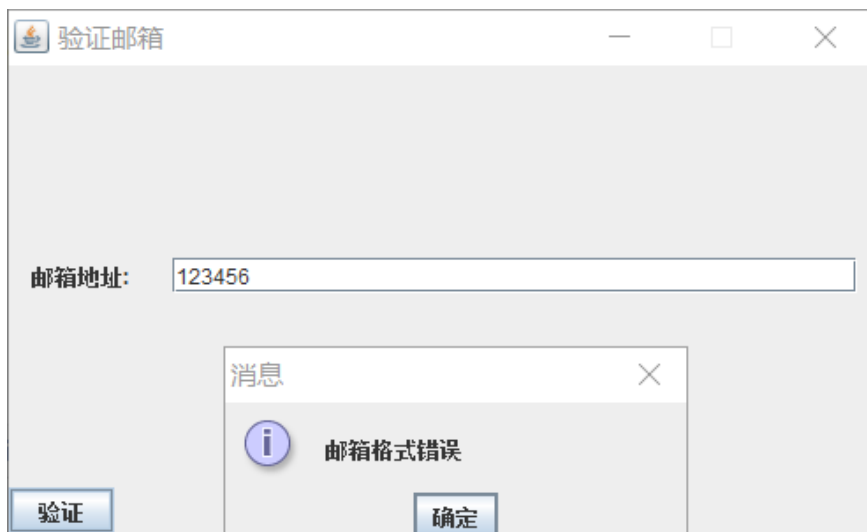


邮件正文: ni hao 2



邮件正文: ni hao 2

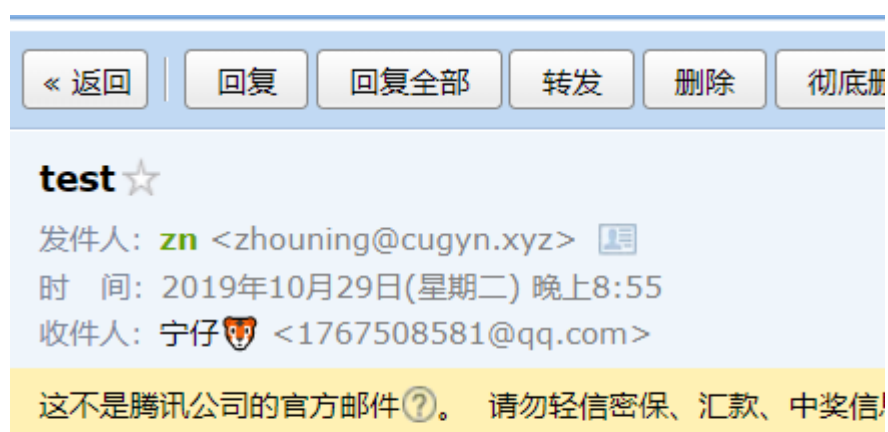
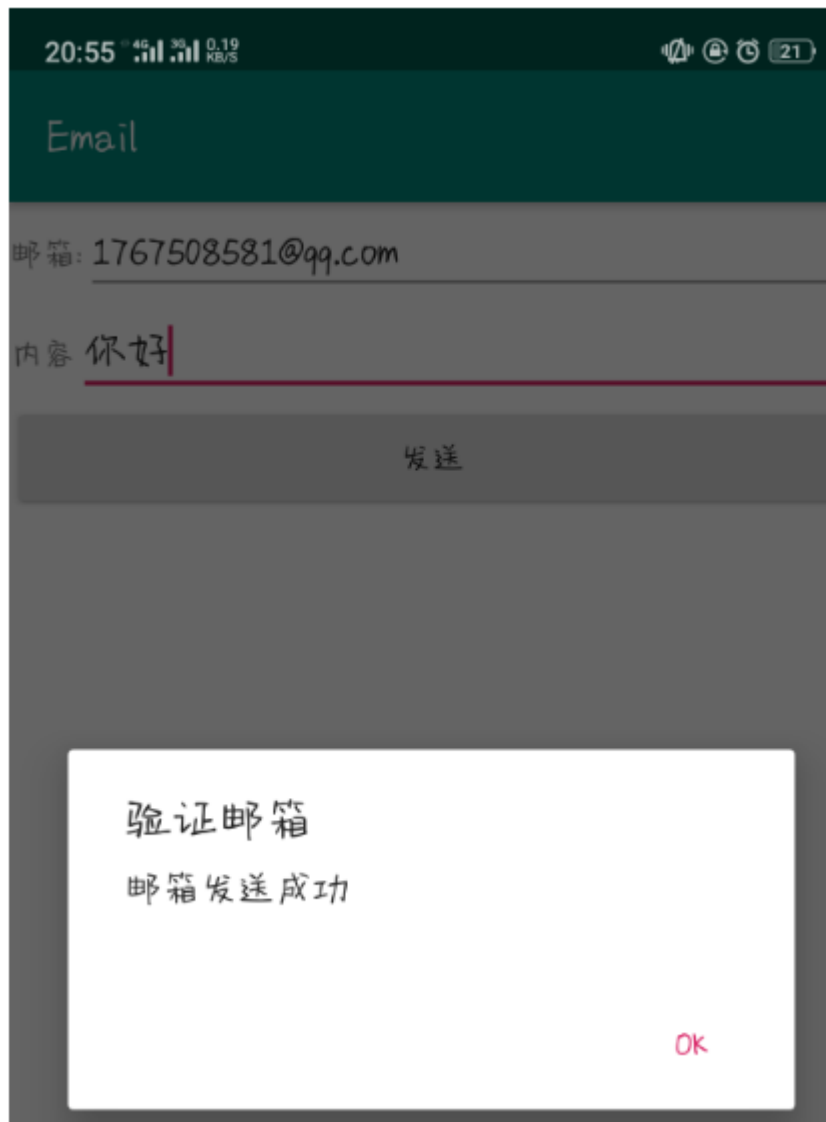
进行邮箱验证：



Android 端起始界面：



发送邮件给自己的 QQ 邮箱



你好

批量发送给自己的 QQ 邮箱和地大邮箱



**test** 🚩 🏠 ⌚ 🖨

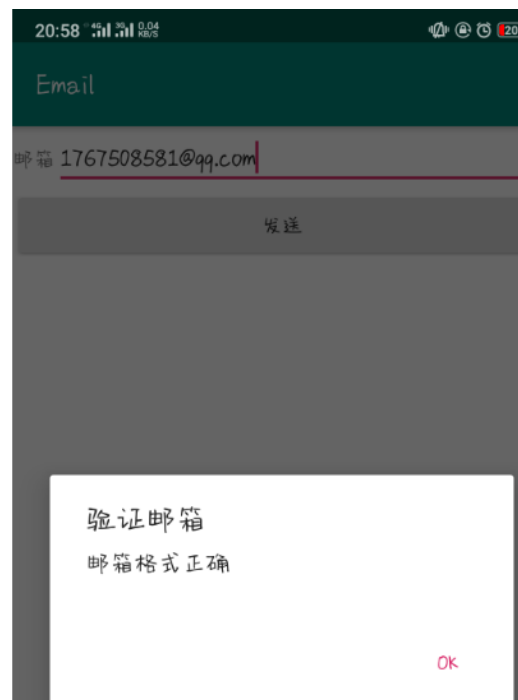
发件人: zn<zhouning@cugyn.xyz> +

收件人: 我<20171004140@cug.edu.cn> +

时 间: 2019年10月29日 20:57 (星期二)

你好2

进行邮件格式识别:



再 AWS 的 EC2 上运行结果

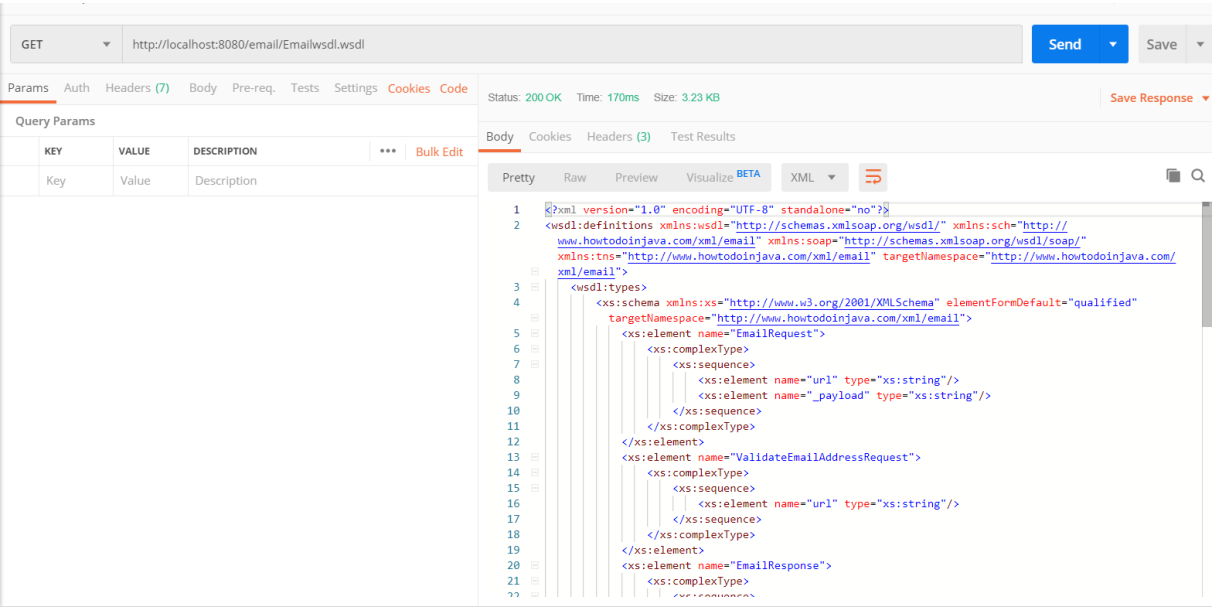
```
ubuntu@ip-172-31-35-251:~/javaProject$ java -jar emailrest-0.0.1-SNAPSHOT.jar

  ____ _
 / ___ \| | | |
/ /___ \| |_| |
 \___ \|____|_|_|
   ____ _
  / ___ \| | | |
 / /___ \| |_| |
 \___ \|____|_|_|

:: Spring Boot :: (v2.2.0.RELEASE)

2019-10-29 12:54:09.615 INFO 2590 --- [main] c.e.emailrest.EmailrestApplication
/home/ubuntu/javaProject/emailrest-0.0.1-SNAPSHOT.jar started by ubuntu in /home/ubuntu/javaProject
2019-10-29 12:54:09.622 INFO 2590 --- [main] c.e.emailrest.EmailrestApplication
2019-10-29 12:54:12.120 INFO 2590 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer
2019-10-29 12:54:12.147 INFO 2590 --- [main] o.apache.catalina.core.StandardService
2019-10-29 12:54:12.152 INFO 2590 --- [main] org.apache.catalina.core.StandardEngine
2019-10-29 12:54:12.301 INFO 2590 --- [main] o.a.c.c.C.[Tomcat].[localhost].[/]
2019-10-29 12:54:12.301 INFO 2590 --- [main] o.s.web.context.ContextLoader
2019-10-29 12:54:13.610 INFO 2590 --- [main] o.s.s.concurrent.ThreadPoolTaskExecutor
2019-10-29 12:54:14.023 INFO 2590 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer
2019-10-29 12:54:14.030 INFO 2590 --- [main] c.e.emailrest.EmailrestApplication
2019-10-29 12:55:20.819 INFO 2590 --- [nio-8080-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/]
2019-10-29 12:55:20.819 INFO 2590 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet
2019-10-29 12:55:20.833 INFO 2590 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet
email服务被调用: 1767508581@qq.com: 你好
validateEmail服务被调用: 123458769
false
```

访问基于 soap 的 webservice 的 wsdl



4.2 结果分析

[对本项工作进行分析评价：说明所得到的结果是否达到预期目标，若可能给出对结果的说明、解释或猜测，给出实验的局限性以及这些限制对结果的影响等。]

结果基本符合预期的期望，但是未给程序进行大量的测试，如同时发送大量邮件等，可能还会存在着许多问题。

五、总结

如：

- (1) [对本次实验报告工作的整体总结。包括：结果要点和结论，本次实习收获和不足。]
- (2) [给出开发中遇到的问题及解决办法，程序中待解决的问题及改进方向。]
- (3) [总结性地阐述本实习结果可能的应用前景、局限性及需要进一步深入研究的方向。]

本次实习是对 WebService 实习的一次锻炼，练习了上课时讲述的 webservice 的使用，再实习当中我学习了我学习了 soap 消息的编写，WSDL 描述以及 xsd 文件的编写，还对 springboot 的使用学习了一下，并且使用了不同的客户端对 webservice 进行访问，体会到其优越的便利性，最终将其放到 aws 的 EC2 上，还实现了真正的发布服务，总体来说时收获不少，不足是自己对 springboot 以及 webservice 没有进行深入的学习，希望再之后学习的过程中继续努力，将其学习的更加深刻。