《软件体系结构与设计》实习报告

题目： 第 3 次上机实验

班级序号： 111171

学生姓名： 董安宁

任课教师： 尚建嘎

中国地质大学信息工程学院软件工程系

2019 年 10 月

一、引言

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

实验地点：未来城校区公教 2-503

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

技术环境需求：

服务器需要 java JDK 1.8 环境，能够运行 Spring 架构打包好的 jar 包。

功能需求：

完成基于 Android，EXE，HTML 的邮件发送服务，邮箱地址查阅服务

客户端需要实现：

① 拥有交互性好的操作界面

② 提供输入，查询功能的接口

③ 提供网络服务，与服务器的接口形成消息交互。

服务器需要实现：

① REST 和 SOAP 的接口

② 每个接口对应的邮件发送和邮箱地址查询的功能。

质量需求：

无论执行成功与否，都应该给出相应的反馈，程序不会因此意外终止。

约束：

使用基于 REST 和基于 SOAP 两种风格的接口完成上述和任务。

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]

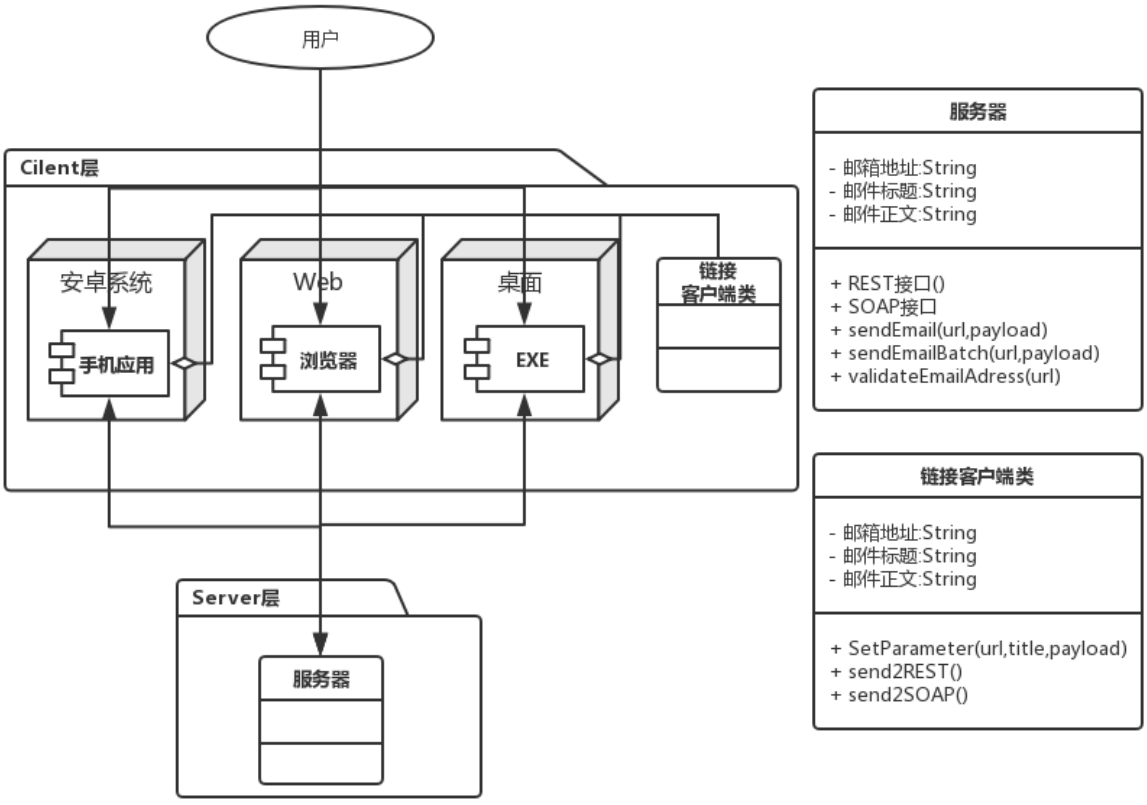

图 1 软件架构图

软件架构如图所示，我采用了 CS 架构来完成该程序，其思想是分别部署服务器和客户端，服务器部分是多个用户共享的信息与功能，执行后台服务，如控制共享数据库的操作等;客户机部分为用户所专有，负责执行前台功能，在出错提示、在线帮助等方面都有强大的功能，并且可以在子程序间自由切换。由于服务器以及由 Spring 架构提供了 REST 和 SOAP 的接口，不管是什么客户端，只需要依照要求访问该接口，就可以执行相应的处理任务，并得到返回值报文。显示给用户。

封装向客户端发送消息的类，提供传递参数和分别表用 REST 和 SOAP 接口的功能，客户端的每个程序都会调用该类进行发送。

其中做出主要处理的是 Server 层的服务器类，该构件的主要功能有

① sendEmail 和 sendEmailBarch

提供了单独发送邮件和群发的功能，返回值为 Y 或 N，表示功能执行的状态。

② validateEmail

提供了判断邮箱地址是否符合要求的功能，返回值为 Y 或 N，表示是或否。

③ REST 接口

提供给外部访问上述服务的接口，通过唯一标识的地址访问服务器，通过传入指定路径和特定格式的参数来访问对应地址的接口函数，在接口函数中调用上述功能，在服务器进行处理后将结果返回。

④ SOAP 接口

提供给外部访问上述服务的接口，通过设置 EndPoint 作为唯一标识地址，客户端封装指定格式的 xml 文件发送到该地址来访问接口函数，在接口函数中调用上述功能，在服务器处理后将结果以 XML 格式返回。

2.3 接口设计

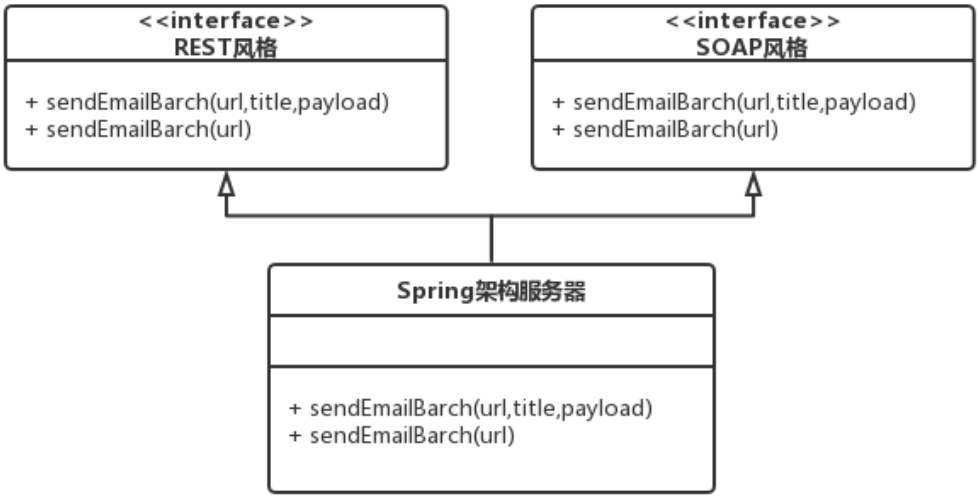

图 2 接口设计

可以看到只有两个接口，分别是 REST 风格接口和 SOAP 风格接口，服务器里的 Spring 架构分别实现 REST 风格和 SOAP 风格的接口，使得访问该服务器的客户端都可以选择调用相应接口。

三、实验过程

3.1 软件实现

本系统使用了 Spring 架构进行开发。REST 和 SOAP 风格的的框架已经给出，在 maven 中设置相应框架的包就可以进行安装部署，当框架部署完毕后再实现具体的接口调用即可完成。开发时使用了迭代递增的方式，每次都只完成某一部分的构件或功能，测试无误后，再在这个基础上扩展功能或者组合功能，再进行测试，再进行后续开发，以此类推。最终一步步搭建出总体的系统。

3.2 实验环境

处理器： i7-7700HQ	操作系统：windows10
开发语言：java	服务器：AWS
实验场景：宿舍，教室	IDE 使用：IntellJ IDEA

3.3 实验步骤

① 安装配置 Spring 环境

第一步首先是安装配置环境，首先我下载了 IntellJ IDEA，在新建项目的时候选择 Spring Initializr，此时接下来的项目就会从 Spring 的官网下载框架包，我们就可以在该框架下部署 REST 风格和 SOAP 风格的服务了。

REST 风格和 SOAP 风格的选择也在新建项目的时候完成，选择如下图所示的两个选项，就会在生成项目的时候自动配置这两种风格。

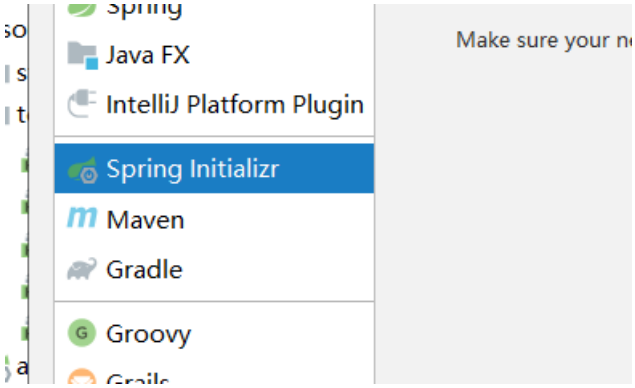

图 3 选择 Spring 框架

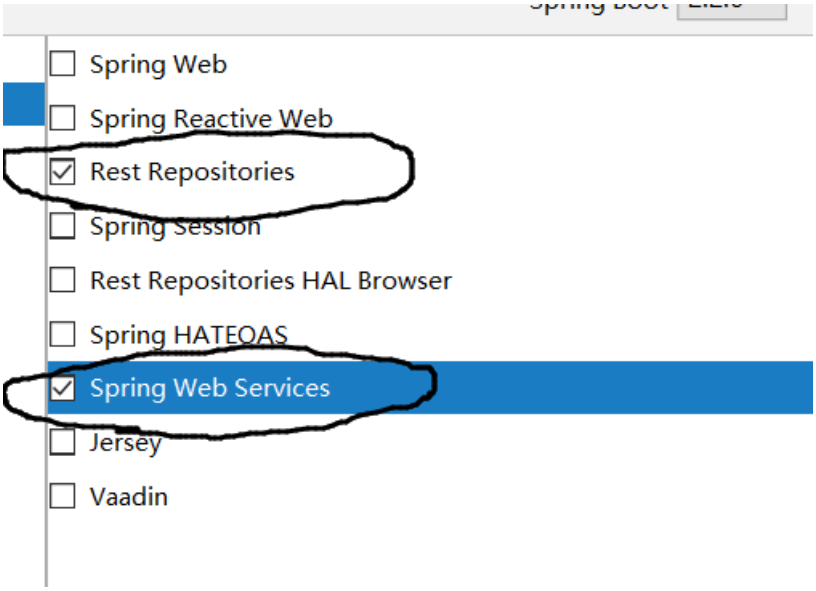

图 4 选择 REST 和 SOAP 风格

等项目建立起来后，就可以使用 maven 配置阿里云的依赖包，方法是把 dependence 部署在 pom.xml 里，如右图所示，等待下载完成后，就可以使用邮件发送的函数了。

```
<dependency>
    <groupId>com.aliyun</groupId>
    <artifactId>aliyun-java-sdk-core</artifactId>
    <version>3.0.0</version>
</dependency>
<dependency>
    <groupId>com.aliyun</groupId>
    <artifactId>aliyun-java-sdk-dm</artifactId>
    <version>3.1.0</version>
</dependency>
```

图 5 maven 配置阿里云的依赖包

当环境配置完毕后进行实验，使用官网教程提供的教程[1]发送邮件。在发送邮件之前首先需要将邮件服务和自己的域名绑定，在这里参考了博客教程[2]，首先购买了一个自己的域名，然后解析该域名，与阿里云邮箱服务的要求同步，然后就可以使用阿里云邮箱服务了。

② 测试使用 REST 风格接口

REST 是一种软件架构风格，或者说是一种规范，其强调 HTTP 应当以资源为中心，并且规范了 URI 的风格；规范了 HTTP 请求动作（GET/PUT/POST/DELETE/HEAD/OPTIONS）的使用，具有对应的语义。

参考搭建 REST 风格的教程[3]我在项目中设置了 REST 风格的接口，其本质是用@SpringBootApplication 标记服务器运行的转函数；用@RestController 标记 REST 风格接口；用@RequestMapping 来标识 URL 中的资源访问，根据输入不同的资源来调用不同的函数；用@RequestParam 标记传入的参数，依此提高函数处理的功能。在有了以上的知识储备后就可以进行 REST 风格的基本测试了。

首先设置传入参数为 url，title，payload，分别代表邮箱地址，标题，邮件正文。如下图所示：

```
//发送邮件
@RequestMapping(value = "/send")
public String GetInfoMation(@RequestParam(name = "url") String url,//邮箱地址
                            @RequestParam(name = "title") String title,//文件标题
                            @RequestParam(name = "payload") String payload)  {//输入内容
```

图 6 REST 风格参数设置

在此基础上写了自己的 HTML 页面，也是通过 Rest 的方式访问页面资源，效果如右图所示：

点击按钮后会将参数打包成 POST 请求发送给/send 接口，然后在该接口内执行发送邮件的操作。邮件发送成成功，如下图所示：

图 7 浏览器测试 Rest 风格

③ 测试使用 SOAP 风格接口

SOAP 是简单对象访问协议，是一种轻量的、简单的、基于 xml 的远程访问协议。可以与现有的多种传输层或应用层协议结合使用，如 TCP、HTTP、SMTP 等。SOAP 广泛使用的是基于 HTTP 和 xml 协议的实现（SOAP=RPC+HTTP+XML），也就是大家常提的 Web Service 使用的通信协议。一个 SOAP 方法可以简单地看作遵循 SOAP 编码规则的 HTTP 请求和响应。

SOAP 风格的接口实现比 REST 复杂一些，参考了 CSDN 的教程[4]搭建了基本的 SOAP 框架。SOAP 接口的部署分为如下步骤：

1. 建立控制 xml 格式的 xsd 文件，在其中设置请求报文和回复报文都需要哪些参数，不符合要求的报文会被服务器拒绝。
2. 建立 SOAP 的@Configuration，用来标识访 SOAP 接口的资源路径。
3. 建立 EndPoint，是主要的服务器处理模块，用户发送的 XML 会在这里被接收，通过调用 xsd 中定义的函数调用该 request 报文中所携带的邮件地址等，然后调用发送邮件的服务，将结果再打包成 XML 文件作为 response 返回给客户端。

④ 编写客户端

首先封装好发送邮件的类，提供接收 url，title，payload 的函数接口，当客户端获取到他们的值后调用该接口传入字符串，然后再调用封装好的发送给 soap 和发送给 rest 的函数，就可以实现邮件发送的功能了。该封装类可以在所有客户端中调用。

1 编写桌面端应用

桌面端应用即用 java 写的界面程序，提供邮件发送服务。界面如下图所示

图 8 桌面版邮件发送

其中每个按钮对应的功能如字面所示，分别调用的是 Rest 风格发送，Soap 风格发送，以及测试邮箱地址的格式正确性，请求会通过 HTTP 协议发送到服务器。

2 编写安卓端应用

安卓端的界面如下图所示：

图 9 安卓端邮件发送

和桌面端的应用一样，都分为三个功能按钮和输入界面，不过安卓端可以群发。

3 HTML 页面应用

HTML 页面的展示已经在最上面展示过，这里不多赘述。

⑤ 上传服务器

上传打包 jar 包到服务器。首先要在服务器端安装 java 环境[5]，然后将在本地打包的 jar 文件上传到服务器运行，然后运行部署在本地的客户端远程调用，可以看到接收和相应的结果，如下图所示。

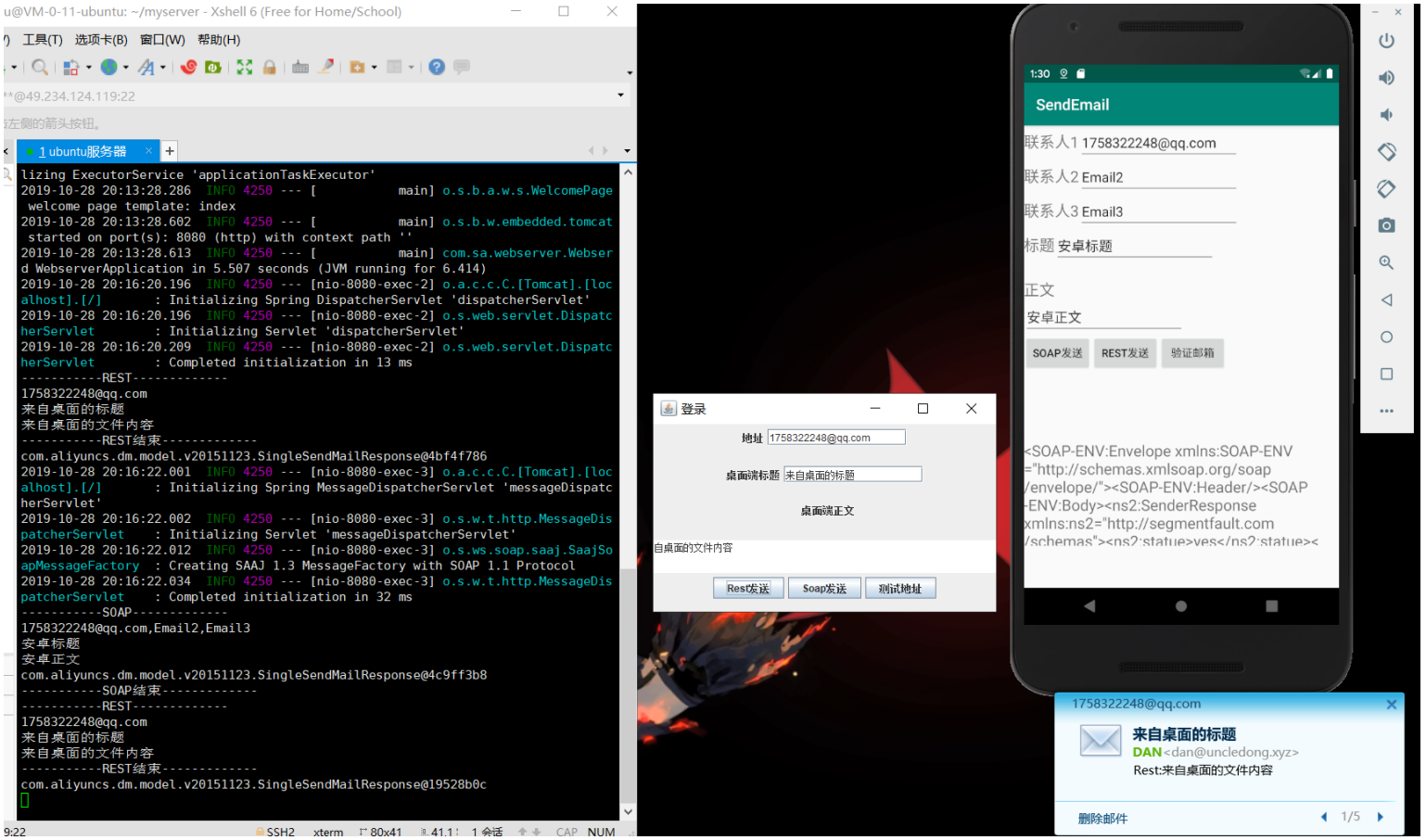

图 10 最终结果

四、实验评价

4.1 实验结果

实验结果在上面进行了给出，我认为较为完整地实现了 REST 和 SOAP 风格接口调用的功能。并且实现了桌面，安卓，web 三种终端的客户端部署。同时也封装了功能和消息的传递，减少了构件之间的耦合程度，通过数据的传输串联该 C/S 架构。

4.2 结果分析

本次实习从结果看我认为很好地实现了 REST 和 SOAP 接口的功能。在这里我采用了 C/S 的结构和 Spring 的框架。Spring 框架为开发提供了一系列的解决方案，比如利用控制反转的核心特性，并通过依赖注入实现控制反转来实现管理对象生命周期容器化，利用面向切面编程进行声明式的事务管理，整合多种持久化技术管理数据访问，提供大量优秀的 Web 框架方便开发等等。因此很轻松就能够部署上述风格。

五、总结

（1）本次实习完成了 REST 和 SOAP 风格的接口，在这个过程中我和很多同学进行了讨论…也学习到了不少东西。本次实习的收获就是使用 maven 添加依赖，使用 Spring 框架，以及两种接口的调用，我发现很多调用形式在日常生活中的网站里经常用到，像我们访问资源地址的时候可能就是用到了其中的某一种风格，这在我们以后的界面开发中提供了更多的选择。

（2）开发中遇到的问题有两个，这也是我收获比较大的

① IDEA 的国内镜像设置

由于在 IDEA 中配置 maven 的时候需要从网上下载依赖包，而默认的地址是国外的，因此速度会很慢，有时候甚至都下载不了。最终我在网上找到了国内镜像的设置[6]，这样一来就不会因为外网原因导致资源下载失败了。

② Android 应用链接网络

根据 StackOverflow[7]上面的回答，我了解到 Android 一般在请求网络服务的时候都是开线程运行，而不能在主线程运行，原因是网络访问有可能会长时间没有访问而使得程序变成未响应状态，这样会导致系统杀死该进程，当然这不是我们希望看到的。因此使用 AsyncTask 类来开线程链接。

（3）本实习的内容可以运用在搭建网站上，运用 Spring 的框架进行资源分配，能够很便捷的提供服务接口，让人把更多精力投入到代码和后端的编写中。以后在做一些成果展示的时候就可以用这种方式了。

参考文献：

参考如下格式列出本次实验所参考的论文、资料的目录

[1] 阿里云邮件服务官方文档：https://help.aliyun.com/document_detail/29459.html

[2] 域名解析和邮件发送：https://blog.janking.cn/post/aliyun-email.html

[3] REST 风格教程 https://www.cnblogs.com/wmyskxz/p/9010832.html

[4] Soap 风格教程 https://blog.csdn.net/panchang199266/article/details/83116941

[5] ubuntu 搭建 java 环境：https://blog.csdn.net/smile_from_2015/article/details/80056297

[6] IDEA 国内镜像：https://www.cnblogs.com/phpdragon/p/7216626.html

[7] StackOverFlow 问题回答：
https://stackoverflow.com/questions/9413625/android-android-os-networkonmainthreadexception

[7]安卓内网络访问：https://developer.android.com/reference/android/os/AsyncTask.html

附件：

代码已经上传到 Github：https://github.com/UncoDong/SAInCug