

学号 20164553

密级

东北大学本科毕业论文

计量管理系统之 微信公众号的开发

学 院 名 称 : 计算机科学与工程学院

专 业 名 称 : 计算机科学与技术

学 生 姓 名 : 施伯贻

指 导 教 师 : 高福祥 教授

二〇二〇年五月

郑 重 声 明

本人呈交的学位论文，是在导师的指导下，独立进行研究工作所取得的成果，所有数据、图片资料真实可靠。尽我所知，除文中已经注明引用的内容外，本学位论文的研究成果不包含他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确的方式标明。本学位论文的知识产权归属于培养单位。

本人签名： 施伯贻 日期： 2020年6月10日

摘 要

计量管理系统是为 CNAS 实验室校准业务所开发的一个款线上软件，运用互联网和云平台技术，实现校准业务的信息化管理。目前该桌面软件的应用已经趋于成熟，为了扩展业务，需要在手机等移动设备上业务延伸和功能映射。让更多的用户了解校准实验室，并能更方便的在移动设备上自助查询，因此选择了拥有庞大用户量作为基础的微信公众平台作为开发平台。

开发此微信公众号用作推广宣传校准实验室以及移动端的部分功能实现。为开发微信公众号需要以 web 工程作为开发项目，开发工具为 eclipse IDE，web 应用服务器选择 Tomcat。编程语言为 Java 和 HTML，Java 用于后台控制，HTML 用于前台网页设计。设计实现了机构简介，新闻推送，查询订单、证书，以及在线意见页面，用户能够在移动端网页和服务器后台交互，并获取到自己需要的信息。此外系统还设置了自定义菜单，自动回复，微信授权等功能，能够在后续需求增加的情况下将这些功能改善加以使用。

本论文是针对开发的这个项目进行的描述。微信公众号是一个系统工程项目，按照软件开发的过程来看，经过了需求分析、系统设计、系统实施和系统测试这些步骤，并且开发过程中，也经历了多次迭代。论文会说明开发过程遇到的问题和解决方案，描述此系统的实现程度以及实用性。

关键字：计量管理系统；微信公众平台；微信公众号；Java

Abstract

Measurement management system is an online software developed for CNAS laboratory calibration business. It USES Internet and cloud platform technology to realize information-based management of calibration business. At present, the application of this desktop software has become mature. In order to expand business, business extension and function mapping need to be carried out on mobile devices such as mobile phones. In order to let more users know about the calibration laboratory and make it more convenient to conduct self-query on mobile devices, WeChat public platform with a large number of users is selected as the development platform.

This WeChat public number is developed to be used as part of the function realization of the promotion and publicity of calibration laboratory and mobile terminal. In order to develop the WeChat public, we need to take web engineering as the development project, eclipse IDE as the development tool, and Tomcat as the web application server. The programming languages are Java and HTML. Java is used for background control and HTML is used for foreground web page design. The design realizes the introduction of the organization, the news push, the inquiry order, the certificate, and the online opinion page. The user can interact with the server in the mobile terminal and get the information they need. In addition, the system also sets up a custom menu, automatic reply, WeChat authorization and other functions, which can be improved and used in the case of increased demand in the future.

This thesis is for the development of this project for a system overview. WeChat public number is a system engineering project. According to the process of software development, it has gone through the steps of demand analysis, system design, system implementation and system test. In addition, it has gone through many iterations in the development process. The paper will explain the problems and solutions encountered in the development process, and describe the degree of implementation and practicability of the system.

Key Words: Measurement management system; WeChat public platform; WeChat public account; Java

目 录

| | |
|-----------------------|-----|
| 郑重声明 | I |
| 摘要 | III |
| Abstract | V |
| 第 1 章 绪论 | 1 |
| 1.1 项目背景与意义 | 1 |
| 1.2 相关研究现状及分析 | 1 |
| 1.3 主要研究内容 | 2 |
| 1.3.1 前台功能 | 2 |
| 1.3.2 后台功能 | 2 |
| 1.4 本文的结构组成 | 2 |
| 第 2 章 相关开发工具介绍 | 3 |
| 2.1 Java | 3 |
| 2.1.1 JDK | 3 |
| 2.1.2 Java EE | 3 |
| 2.2 Tomcat | 3 |
| 2.3 公众号开发者工具 | 4 |
| 2.3.1 开发者文档 | 4 |
| 2.3.2 在线接口调试工具 | 5 |
| 2.3.3 web 开发者工具 | 6 |
| 2.3.4 公众平台测试账号 | 7 |
| 第 3 章 系统分析 | 9 |
| 3.1 系统现状分析 | 9 |
| 3.2 公众号需求分析 | 9 |
| 3.2.1 功能需求分析 | 9 |
| 3.2.2 软硬件需求分析 | 10 |
| 第 4 章 系统设计 | 11 |
| 4.1 总体功能设计 | 11 |
| 4.1.1 自动回复功能 | 12 |

| | |
|--|-----------|
| 4.1.2 自定义菜单功能 | 12 |
| 4.1.3 微信网页设计 | 13 |
| 4.2 数据库设计 | 14 |
| 第 5 章 系统实现..... | 17 |
| 5.1 Java 环境和 Tomcat 的配置 | 17 |
| 5.2 微信公众号注册与选择 | 19 |
| 5.3 开发者模式配置说明 | 20 |
| 5.3.1 URL | 21 |
| 5.3.2 Token | 22 |
| 5.3.3 EncodingAESKey 和消息加解密方式..... | 22 |
| 5.4 开发者模式接入 | 22 |
| 5.5 access_token 的获取 | 23 |
| 5.6 自定义菜单的创建 | 24 |
| 5.7 消息和事件接收 | 27 |
| 5.7.1 普通消息接收 | 27 |
| 5.7.2 接收事件推送 | 28 |
| 5.8 消息和事件的自动回复功能 | 29 |
| 5.8.1 图片消息的回复 | 30 |
| 5.8.2 音乐消息的回复 | 31 |
| 5.8.3 图文消息的回复 | 32 |
| 5.9 微信网页开发 | 33 |
| 5.9.1 查询进度 | 35 |
| 5.9.2 查询证书状态 | 36 |
| 5.9.3 查询仪器 | 37 |
| 5.9.4 在线意见 | 37 |
| 5.10 微信网页授权 | 38 |
| 5.10.1 用户同意授权，获取 code..... | 39 |
| 5.10.2 通过 code 换取网页授权 access_token | 40 |
| 5.10.3 拉取用户信息(需 scope 为 snsapi_userinfo) | 41 |
| 5.11 第三方网站绑定微信账号 | 41 |

第 6 章 系统测试..... 43

6.1 简介新闻 43

6.2 查询 44

6.2.1 查询进度 44

6.2.2 查询证书状态 45

6.2.3 仪器检测 46

6.2.4 错误页跳转 47

6.3 在线意见 47

第 7 章 结论 49

7.1 总结 49

7.2 展望 50

参考文献..... 51

致 谢 53

第1章 绪论

1.1 项目背景与意义

本课题是辽宁省计量科学研究所的计量管理系统平台的增加功能。辽宁省计量科学研究所的主要业务是计量仪器的检定和校准。现在已经上线了计量管理系统平台，LIMS 计量管理平台主要适用于 CNAS 校准实验室，实现校准业务的信息化管理。平台基于校准业务主线，实现了委托合同、现场派单、仪器流转、划价收款、出具证书、审批证书、发放仪器和证书等流程的电子化，既有效地提高了校准工作效率，又为公司的发展决策提供了数据支撑。

随着手机应用的发展，为继续推动“互联网+”计量创新服务模式，进一步方便广大客户获取检测信息、掌握检测进度^[1]，需要在计量管理平台的基础上，扩展在手机等移动设备上使用的功能。该微信公众号是辽宁省计量科学研究所下属的校准公司开发的，亟需开发的微信服务平台的目的包括：

- (1) 做好品牌宣传，便于其客户了解校准实验室；
- (2) 依托计量管理平台，为客户提供一些自助查询服务；

基于此，其功能包括公司介绍，查询认可能力、查询委托单进度、公司新闻推送等。微信公众平台的开通从意义上来看，能够实现研究所与广大客户的“微距离”，提高沟通效率，提升服务质量，是研究所信息化服务计量业务发展的新体现，也是为客户提升服务质量的新举措。这在经济、社会效益层面都具有实际应用价值。

1.2 相关研究现状及分析

就单独对于公众号的研究，目前国内的书籍多是出于运营、营销等角度，因为开发公众号的一大目的就是考虑它的经济效益。如何运营好微信公众号，分享移动互联网的社交红利，成为许多企业研究探索的一个重要课题^[2]。有一份调查指出，微信中的集体关注度与其它社交媒体遵循类似的模式，了解用户的关注度对于广告和营销是非常有用的^[3]。

然而国内对微信公众号的视觉、界面或者 UI 等研究尚且不足^[4]。不过因为公众号这一存在的出现，改变了行业内共享信息的方式，拓宽了学者们的视野，也因此有利于学术研究的发展。作为一个新的平台，微信官方账号是免费的、依赖于互联网的、

易于管理的^[5]，公众号作为一种新兴的重要媒体形式，被用于虚拟团队^[6]的建设和知识共享。通过微信公众号，特定技术领域的发展和成果可以在一些主流机构或组织运营的微信公众号中体现和分享^[7]。

而在传播方面也有一些研究成果，如：微信公众号文章的阅读量与时间满足幂律关系^[8]；用户阅读并转发微信公众号信息的概率随着转发人数增加而增加，而当转发该信息的好友人数达到一定的阈值后，该用户转发该信息的概率反而会减小^[9]；在微信中，用户传播行为与信息吸引力正相关，不同的文章对用户的吸引力是不一样的，文章内容越有价值或越吸引人，信息传播的范围就越广^[10]。

1.3 主要研究内容

1.3.1 前台功能

- (1) 机构：简介、新闻、资质、联系方式。内容以文字+图片描述为主。
- (2) 查询：进度、价格、证书真伪。用户输入委托单号或证书编号后，可在计量管理平台中自动查询出相关信息，反馈给用户。
- (3) 服务：在线投诉及建议。用户可选择匿名/不匿名方式进行投诉，可根据需要上传一些图片。

1.3.2 后台功能

- (1) 完善机构相关信息，包括修改简介、编辑发布新闻等。
- (2) 浏览在线投诉信息，录入投诉处理结果等。

1.4 本文的结构组成

主要内容为第2~6章。

第2章为相关开发工具介绍，介绍此项目应用到的开发工具。

第3章是系统分析，分析企业线上系统现状，分析设计此公众号系统的需求。

第4章是系统设计，介绍该项目总体和各个部分详细设计的过程，还有此过程中所经历的问题解决和指导，经过图表来表达这一方面的内容。

第5章是系统实现，通过截图等描述该公众号所实现的功能，以及通过代码片段和注释来重点强调某些重要部分代码的实现。

第6章是系统测试，通过测试主要功能来测试该公众号的实用性。

第 2 章 相关开发工具介绍

2.1 Java

Java 是一门面向对象编程语言，作为静态面向对象编程语言的代表，极好地实现了面向对象理论，允许程序员以优雅的思维方式进行复杂的编程^[11]。

2.1.1 JDK

JDK (Java Development Kit)是 java 程序开发包，是 java 程序在服务器上运行时必须得装配的环境。它内部包含 Java 所需要的运行环境 JRE(JAVA Runtime Environment)。而本工程需要大量的 jar 包作为工具而不仅限于执行一个简单的 java 程序，因此安装环境选择虽然占据内存但是功能较全的 JDK 作为安装环境。

2.1.2 Java EE

Java EE（Java Platform，Enterprise Edition）是一款企业级 java 开发应用。它是在 Java SE（Java Standard Edition）的基础上进行改进的版本，更适合较大项目的开发，支持公众号开发所需要的的 web 服务。

2.2 Tomcat

目前主流的 web 服务器主要有 Apache、Nginx、IIS。而 tomcat 是基于 Apache 下开发的自由软件，针对的是 Java 语言的 web 应用，支持 servlet 与 JSP 的规范，并且是一款免费的开放源代码的 Web 应用服务器。对于初学者开发是个非常合适的应用，它属于小型的服务器，比较适合用于用户数量较少的情况。如图 2.1 所示为 Tomcat 的主要器件组成图。

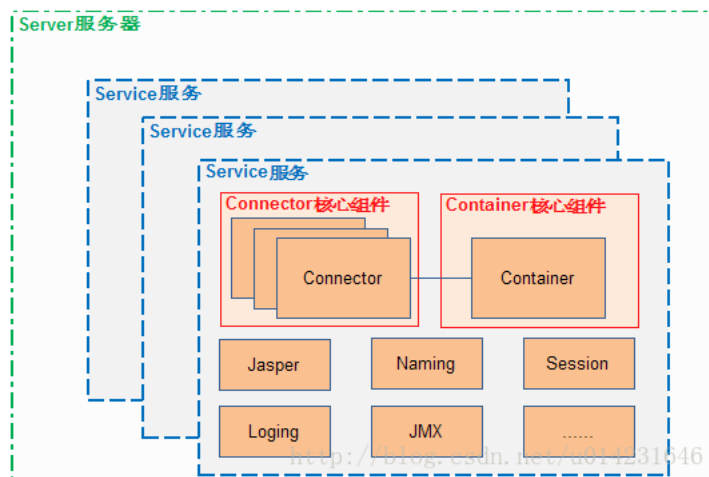


图 2.1 Tomcat 的主要器件组成

其中最重要的是 Connector 和 Container 两个核心组件，他们构成了 Service 服务，而 Service 是组成 Tomcat Server 服务器的基础。当项目启动时，若 Connector 和 Container 报错，即说明连接失败或者容器中出现找不到某个文件的情况。一般是在配置情况下出现的问题。

2.3 公众号开发者工具

如图 2.2 所示为微信提供给开发者用于开发微信公众号的各种官方工具，其中最为重要的是开发者文档，在线接口调试工具，web 开发者工具和公众平台测试账号。



图 2.2 公众号开发者工具

2.3.1 开发者文档

开发者文档是一个在线网页，为公众号开发者开发前必须阅读数遍的参考工具，里面介绍了公众号所有接口调用的权限和方法，如图 2.3 所示：



图 2.3 开发者文档界面

2.3.2 在线接口调试工具

微信公众号本身是一个自主研发的新平台，开发者在开发过程中不免会碰到接口调用失败等问题，往往是因为请求错误，发送错误，或提交参数有问题。通过接口调试工具可以填写自己在获取接口时所使用的参数，检查问题后会返回一个错误码，可以在开发者文档中查找错误码所代表的错误，并根据错误进行代码的修改。如图 2.4 为通过接口调试工具测试 `access_token` 的获取是否出现差错的过程，通过结果可以看出此接口没有出现问题。

微信公众平台接口调试工具

此工具旨在帮助开发者检测调用【微信公众平台开发者API】时发送的请求参数是否正确，提交相关信息后可获得服务器的验证结果

使用说明：

(1) 选择合适的接口。

(2) 系统会生成该接口的参数表，您可以直接在文本框内填入对应的参数值。（红色星号表示该字段必填）

(3) 点击检查问题按钮，即可得到相应的调试信息。

一、接口类型：

基础支持

二、接口列表：

获取access_token接口 /token

 方法：GET

三、参数列表：

* grant_type：

client_credential

获取access_token填写client_credential

校验通过

* appid：

wx6314499fa6d9d51a

填写appid

校验通过

* secret：

e9998498f9ebcbdf1387f5241b02t

填写appsecret

校验通过

检查问题

(a)

基础支持: 获取access_token接口 /token

请求地址：

https://api.weixin.qq.com/cgi-bin/token?grant_type=client_credential&appid=wx6314499fa6d9d51a&secret=e9998498f9ebcbdf1387f5241b02b586

返回结果：

200 OK

Connection%3A%20close
Date%3A%20Sun%2C%2024%20May%202020%2017%3A%20%3A08%20GMT
Content-Type%3A%20application%2Fjson%3B%20encoding%3Dutf-8
Content-Length%3A%20194
false

提示：

Request successful

(b)

图 2.4 在线接口调试工具测试 `access_token` 的获取接口

-5-

2.3.3 web 开发者工具

web 开发者工具是一款由微信官方开发的桌面应用，可以让开发者在 PC 或 Mac 上模拟访问微信内网页，便于开发者进行调试。开发者可以在这款应用上调试微信授权页面，微信网页开发和 JS-SDK 相关功能与权限。它利用集成的 Chrome DevTools 协助开发，更方便开发者调试页面样式设计。图 2.5 为 web 开发者工具的工作界面。



图 2.5 web 开发者工具

！注：为了保证开发者信息安全，需要先在网页登录公众号账号，将开发者微信号与 web 开发者工具进行绑定，图 2.6 为绑定界面。



图 2.6 web 开发者工具绑定页面

2.3.4 公众平台测试账号

正式的公众账号不仅需要企业各种认证，同时部分接口也需要经过工信部备案的域名才可以拥有调用权限，因此不适合用于个人开发。而测试账号则是个人开发最适合的一大开发者工具，它可以不需要开发者申请公众账号就使用一个测试用公众号进行开发，并且能调用所有高级接口，不受域名限制，只需要公网能够进行访问的 ip 地址，只需要经过外网映射即可，图 2.7 为测试号页面。

测试号管理 微信号： gh_f99bb67fe982 退出

测试号信息

appID

appsecret

接口配置信息 [修改](#)

请填写接口配置信息，此信息需要你有自己的服务器资源，填写的URL需要正确响应微信发送的Token验证，请阅读[消息接口使用指南](#)。

URL

Token

JS接口安全域名 [修改](#)

设置JS接口安全域后，通过关注该测试号，开发者即可在该域名下调用微信开放的JS接口，请阅读[微信JSSDK开发文档](#)。

域名

(a)

测试号二维码



请用微信扫描关注测试公众号

用户列表 (最多100个)

| 序号 | 昵称 | 微信号 | 操作 |
|----|-----|-----|--------------------|
| 1 | boy | | 移除 |

模板消息接口

新增测试模板

最多10个，接受模板消息需要关注测试号

| 序号 | 模板ID(用于接口调用) | 模板标题 | 模板内容 | 操作 |
|------|--------------|------|------|----|
| 暂无数据 | | | | |

体验接口权限表

| 类目 | 功能 | 接口 | 每日调用上限/次 | 操作 |
|----|------|--------------------------------|---------------------------|-----|
| | 基础支持 | 获取access_token | 2000 | |
| | | 获取微信服务器IP地址 | 无上限 | |
| | | | 获取设备安全校验码 | 无上限 |

(b)

图 2.7 测试号页面

-7-

第3章 系统分析

3.1 系统现状分析

计量管理平台主要适用于 CNAS 校准实验室开展的校准业务，运用互联网和云平台技术实现校准业务的信息化管理。

平台以符合 CNAS 管理规定、流程简洁、易于操作作为根本宗旨。基于校准业务主线，实现了委托合同、现场派单、仪器流转、划价收款、出具证书、审批证书、发放仪器和证书等流程的电子化。另外，平台还具有权限设置、一键安装、自动升级等功能，较大程度地降低了信息化运维成本。

平台的技术特点和主要优势有以下几个方面：

- (1) 实现了证书模板的集中管理；
- (2) 实现了 Word 和 Excel 版本的电子记录管理；
- (3) 实现了 CNAS 授权认可项目的网上管理；
- (4) 引入电子签名，在网上提交、核验、审批证书，实现证书的全流程网上审批；
- (5) 实现了现场检测派单任务的电子化；
- (6) 实现了客户和联系人的集中统一管理。

3.2 公众号需求分析

3.2.1 功能需求分析

1.3 节说明了研究微信公众平台的前台功能和后台功能，图 3.1 是为实现其前台功能与后台功能而设计的用例图，对于此系统来说，Actor 只有管理员和用户，管理员对于系统要执行后台操作，而用户使用此系统与之交互的界面前台功能，同时用户的三个用例也是我们所需要实现的业务流程。

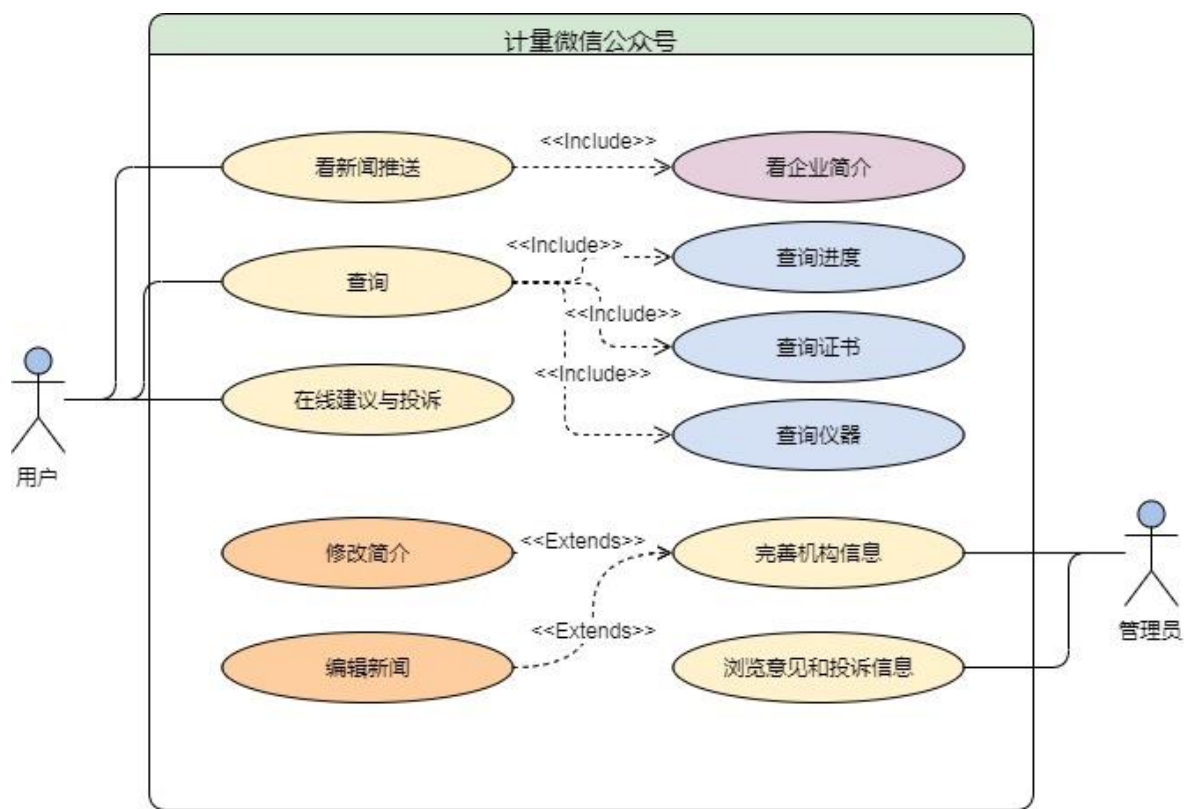


图 3.1 微信公众号系统用例图

3.2.2 软硬件需求分析

根据企业服务号的需要，云服务器作为开发的设备会是一个最佳的选择，通过远程登录云服务器在其之上开发，可以从外网直接访问 IP 地址，而不需要经过外网映射。且作为服务器工程开发也较为方便，本地信息与工程代码分开处理，确保了安全性。

软件方面，从事 Java Web 开发应当选择 eclipse IDE 作为开发软件，Web 应用服务器应该选择 Tomcat，作为小型服务器，适合较为简单的业务交流。同时因为微信网页开发的特殊性，需要下载官方桌面应用 Web 开发者工具用来调试界面样式和测试界面功能等。

第 4 章 系统设计

4.1 总体功能设计

首先要确立的是要设计一个公众号应该分为哪几个步骤。此处默认已开启开发者模式，开发者模式的接入见第 5 章。

第一步需要让微信公众号界面成为一个能与用户交互的界面，若没有经过任何处理的界面只能一味地接收用户发送的消息而不会作出任何回应。内部需要具体实现的是响应用户发送的消息进行自动回复，以及通过设置自定义菜单让用户使用。

第二步是建立能够实现计量微信公众号查询、新闻推送、在线意见的界面，这些功能无法简单地通过聊天界面的消息交互来实现，并且不够直观，应该开发相应页面来处理这些业务逻辑。

第三步是连接接口，将成型的微信公众号界面中能够进行交互的接口与完成功能的页面进行接口连接，因为微信公众号界面中的大部分接口官方是已经建立好的，我们需要的是将它接口连接到正确的路径上。

如图 4.1 为总体设计的包图展示：

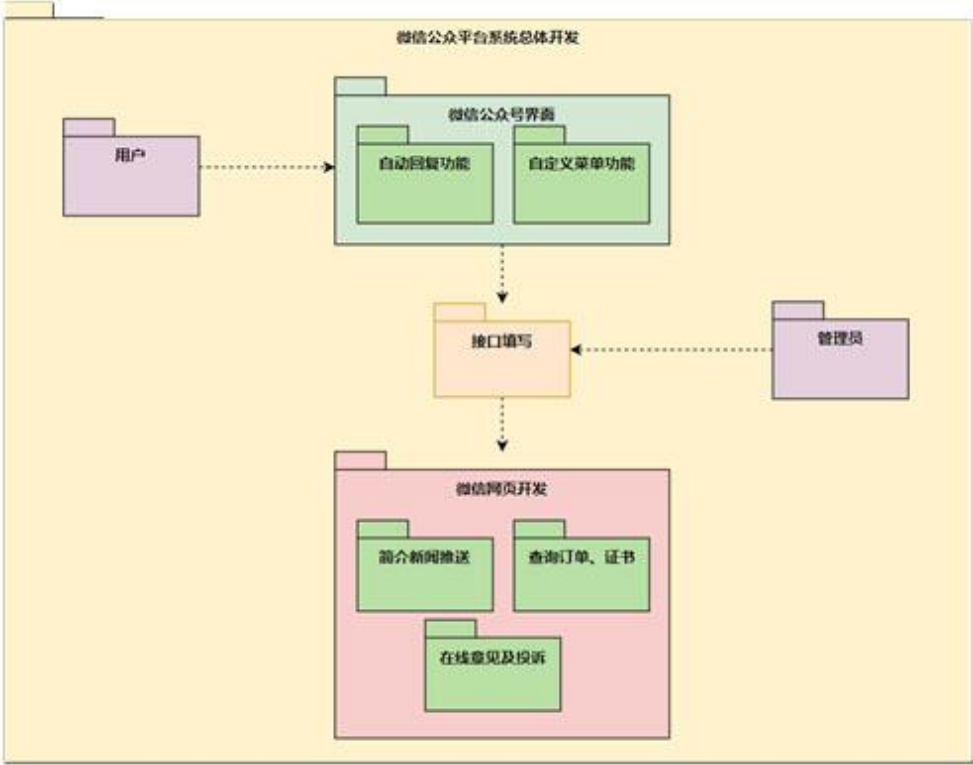


图 4.1 微信公众号总体设计包图

因为接口只需要填写便能够使用，因此详细讨论，公众号界面与微信网页的设计。

4.1.1 自动回复功能

自动回复功能是本项目的附属功能,为了实现公众号聊天主界面的功能完善和界面美观而设计的功能,因此响应的大部分消息均为测试用消息,当后续开发需要时可填写相应的关键字进行改变。

首先用户对页面的操作包括通过发送消息和点击自定义菜单,当微信服务器接收到用户的操作,将操作的信息发送给本地服务器后台,后台会根据发送来的信息进行处理。

其次,根据发送消息的不同,选择回复的消息类型和内容也不同,根据需要,图 4.2 为本项目所需要的消息类型。

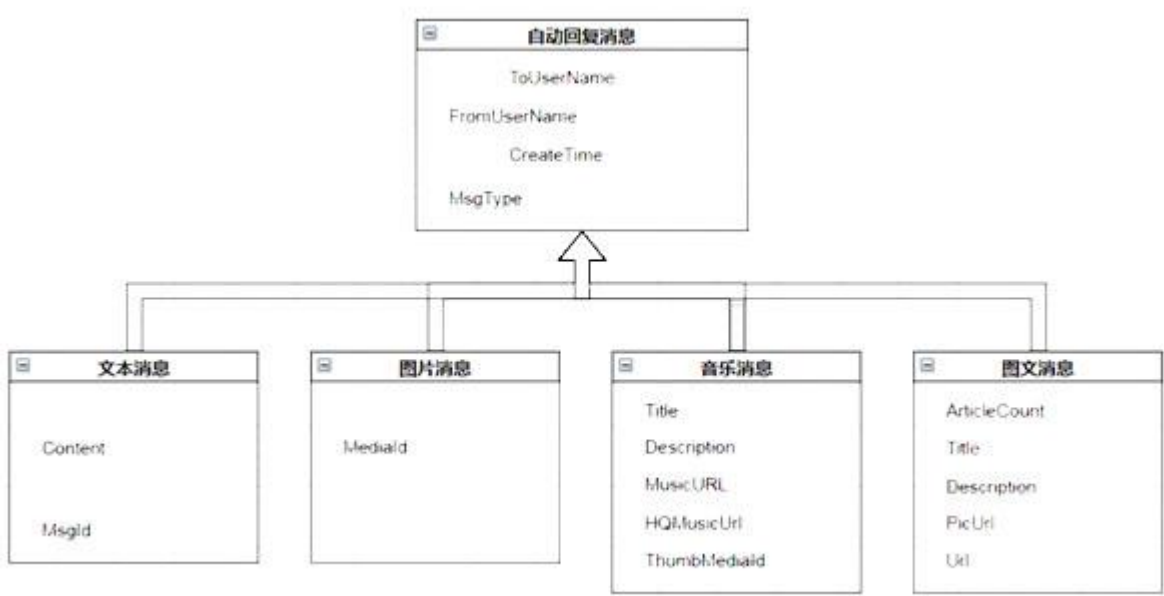


图 4.2 不同的消息类型及其属性

将回复的消息发送回到微信服务器上,微信服务器再检验此消息格式是否正确,格式正确的消息包发送到微信公众号客户端上,呈现出自动回复的内容。

其中本地服务器需要实现的功能为将收到的 xml 数据包转化为可处理的数据,根据消息内容或者事件 Key 值决定回复的内容,并封装信息,将信息转为 XML 数据包后发送回微信服务器上。

4.1.2 自定义菜单功能

目前的业务功能主要在微信网页端,所以通过自定义菜单的按钮跳转到微信网页进行业务处理是本项目非常重要的一个部分。

首先本地服务器需要从微信服务器获取微信公众号接口调用凭证 `access_token`，只有获取了 `access_token` 才能调用创建菜单的接口。

其次根据需要创建按钮和菜单类，并封装菜单栏。按钮和菜单类如图 4.3 所示。

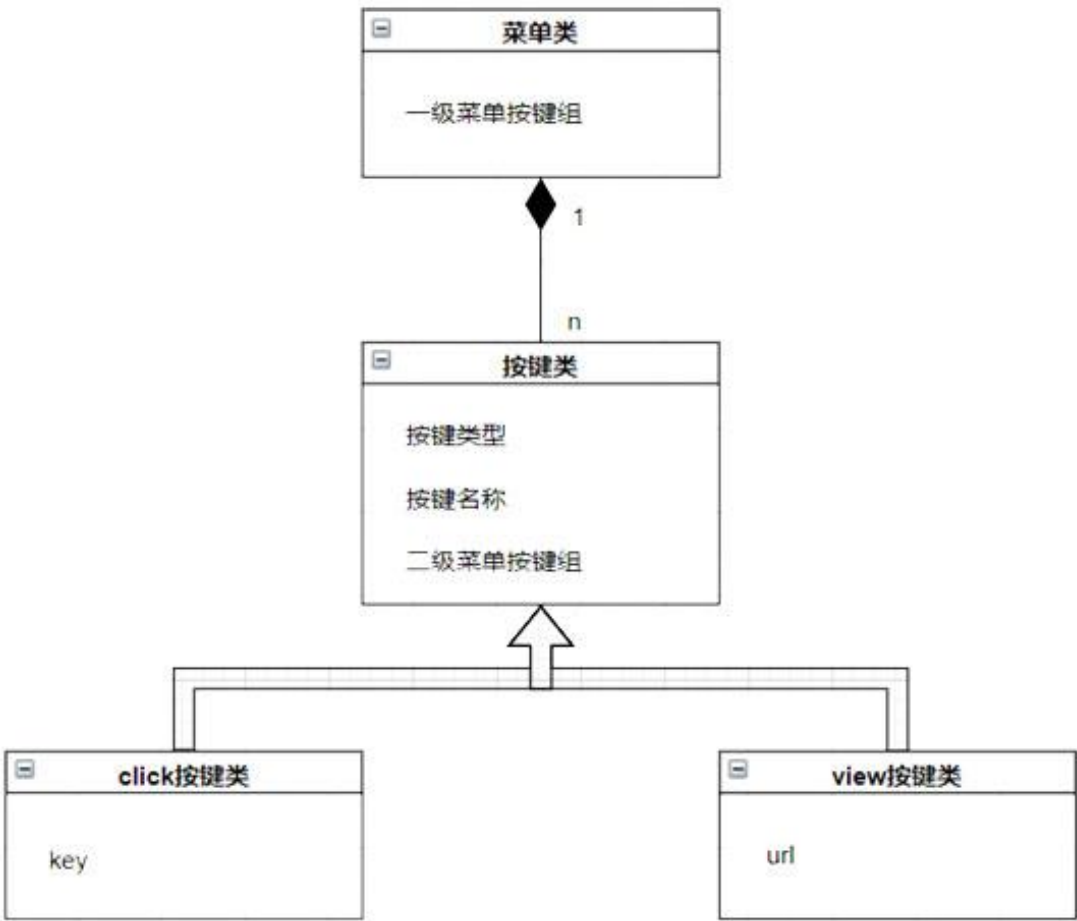


图 4.3 按钮和菜单类

将封装好的菜单栏转化为 JSON 数据包的格式和事先获取到的 `access_token` 发送到微信服务器指定的 URL 下。

当微信服务器收到该数据包后会判断其格式内容是否正确。若正确，在微信客户端创造菜单并且返回的 JSON 数据包会提示菜单创建成功；若失败，则只会返回 JSON 数据包，其中包含错误码和提示菜单创建失败。

其中本地服务器需要实现的功能有获取 `access_token`，创建菜单和按钮类并封装，将封装好的菜单转为 JSON 数据包并上传到微信服务器上。

4.1.3 微信网页设计

此部分为本项目业务实现的重点，尤其是查询功能。因为简介和新闻推送功能都

是简单的 html 界面，没有经过任何后台处理，修改只需要在 jsp 文件里进行补充即可，在这里不多做介绍，以后面系统实现来体现。本节重点说明查询功能的设计过程，因为三个查询的设计过程基本相同，只是其中查询的变量，查询的表不同，只是进行拷贝和简单的替换即可。

其中本地服务器需要实现的功能编写查询、查询成功和查询失败三个页面，能从查询页面获取用户输入信息并封装在 sql 语句中，连接到 oracle 数据并执行 sql 语句进行查询，执行查询结果的判断和页面跳转，跳转的页面将查询到的结果显示出来，如图 4.4 所示为各个页面关系组成的类图。

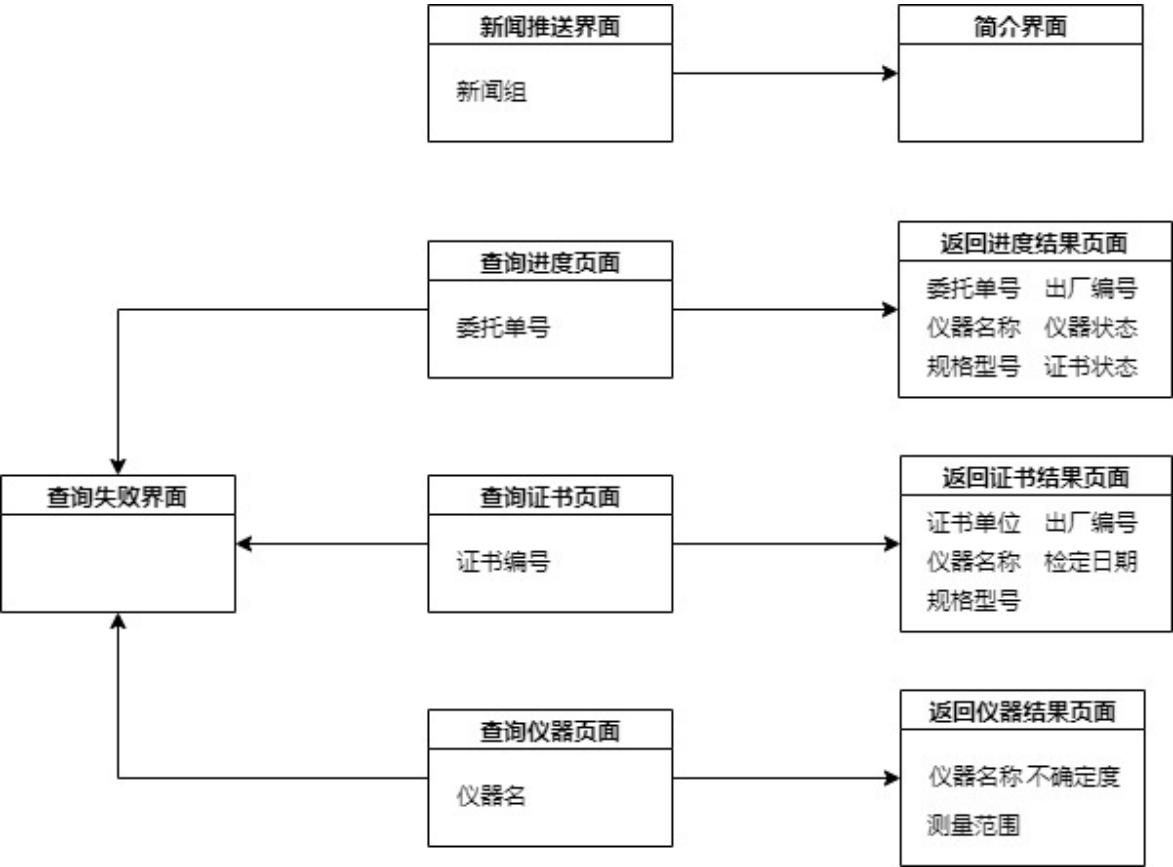


图 4.4 页面类

4.2 数据库设计

计量管理平台线上平台已经营运了多年，有自己的数据库，本项目在查询过程中直接调用其数据库中的表进行查询即可。但是用户类并没有 openid 属性用于储存经过微信授权用户的唯一用户标识，所以仍然需要创建表添加此属性完善功能。如图 4.5 所示为本项目中设计到的数据库中的 E-R 图。

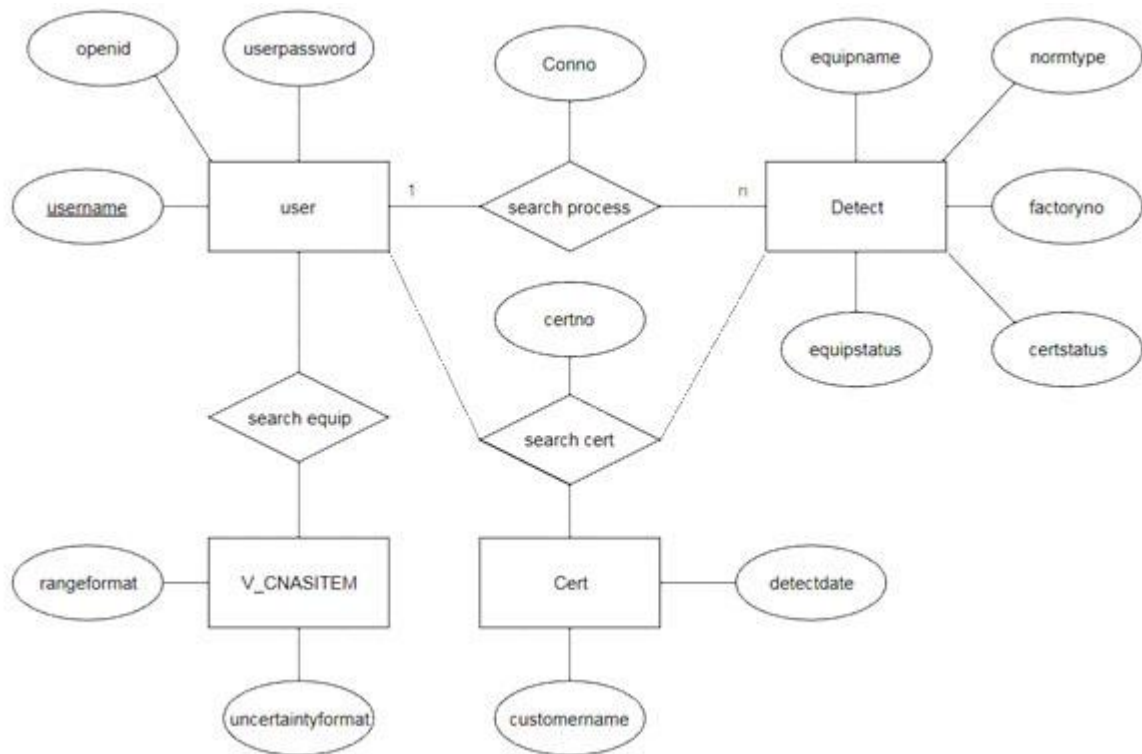


图 4.5 数据库的 E-R 图组成

其中值得注意的是，连接 oracle 数据库也是一个复杂的操作，需要先在 tomcat 服务器上运行代码加载数据库驱动，加载成功后才能够连接数据库。

而在工程中连接 oracle 数据库时需要注意的是，对于连接数据库这一功能，需要创造一个类来实现它的功能，这既是数据库设计的一部分，也是功能设计的一部分。

类中需要四个成员变量：driver,url,username,password。

为了避免以后数据库更换或者使用了另一种非 oracle 的数据库，在加载驱动（driver）时不能把固定值指向 oracle 数据库驱动，应当设立该变量以备不时之需。

因为连接数据库既有连接本地数据库也有连接远程数据库的情况，尤其是以该项目中的情况，需要远程连接数据库，这也是设立 url 成员变量的原因。

不仅仅是要多注明一条 url 的问题，同时还要确保远程数据库是否能在公网上访问，且是否有权访问。若无法确保访问到，那么就无法连接上该数据库所在的服务器端。

另一方面同一个数据库，不同的用户所创建的表是不互通的，像微信授权更新数据库信息和查询委托订单这两个功能所涉及的表为不同用户所创造，因此类中同样要拥有成员变量 username 和 password，代表该数据库的不同用户。

如图 4.6 为 jdbc 类，用以处理连接数据库事务的类。

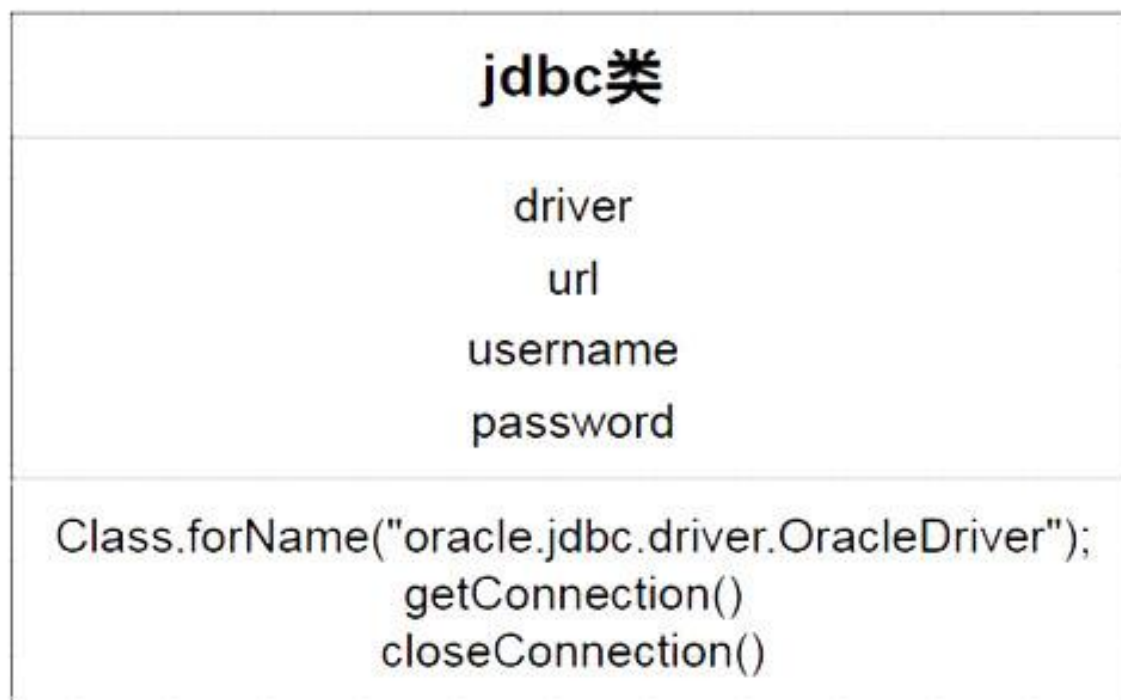


图 4.6 jdbc 类组成

其中 Class.forName("oracle.jdbc.driver.OracleDriver")、getConnection()、closeConnection()为实现加载数据库驱动，连接数据库，关闭数据库连接的三个函数，是该类所实现的功能。

第 5 章 系统实现

5.1 Java 环境和 Tomcat 的配置

本项目所使用的 JAVA 版本为 jdk-13.0.2，在安装后需要在系统高级设置中的环境变量中进行路径配置，如图 5.1 所示，在 Path 中添加此版本所在路径，并多建立变量 JAVA_HOME，值仍然为该软件安装地址。

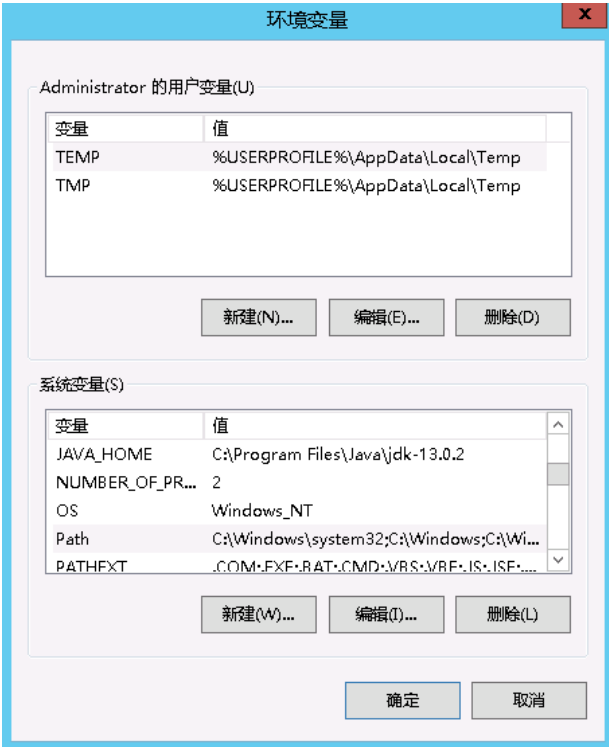


图 5.1 Java 环境的配置

之后在 eclipse 中点击 Window 菜单栏中的 Preferences，在如图 5.2 中的界面配置能够找到 Java 环境变量并在 eclipse 上进行配置。

Tomcat 的配置同理，本设计使用的 Tomcat 版本为 apache-tomcat-9.0.34，因为 10 以上版本可能存在于当前 jdk 版本不兼容的情况。如图 5.3 所示，将 CATALINA_HOME 变量赋值 Tomcat 安装路径，同时添加新变量 TOMCAT_HOME，值也同为路径。

之后同样在 eclipse 中点击 Window 菜单栏中的 Preferences，在如图 5.4 中的界面配置能够在 eclipse 上搜索到 Tomcat 并使用。

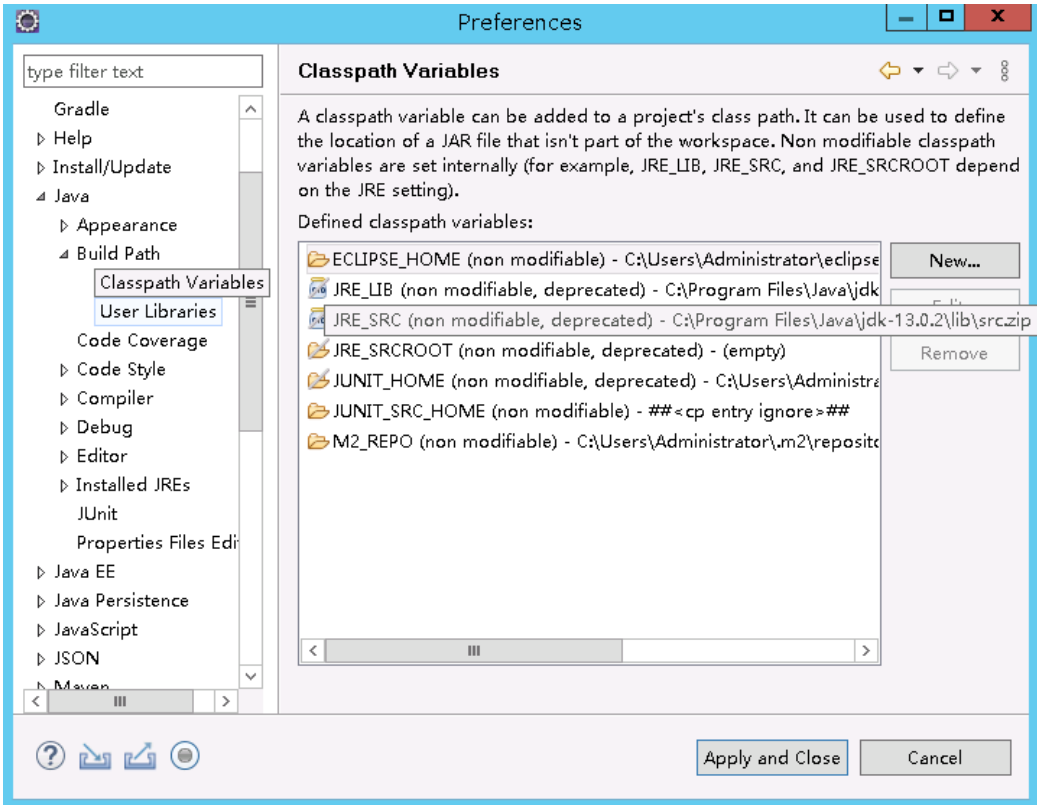
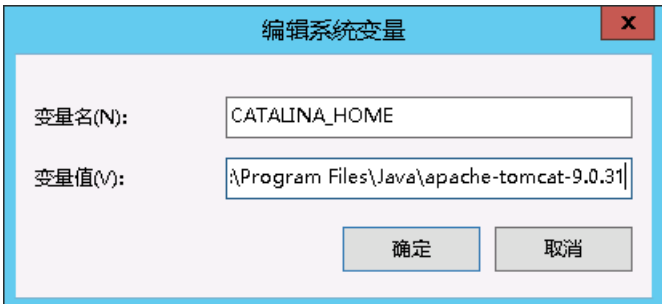
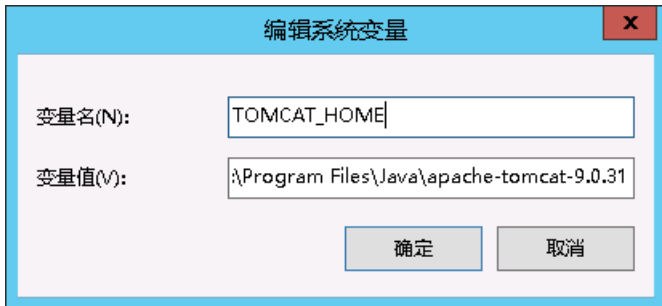


图 5.2 eclipse 中配置 java 环境



(a)



(b)

图 5.3 Tomcat 环境的配置

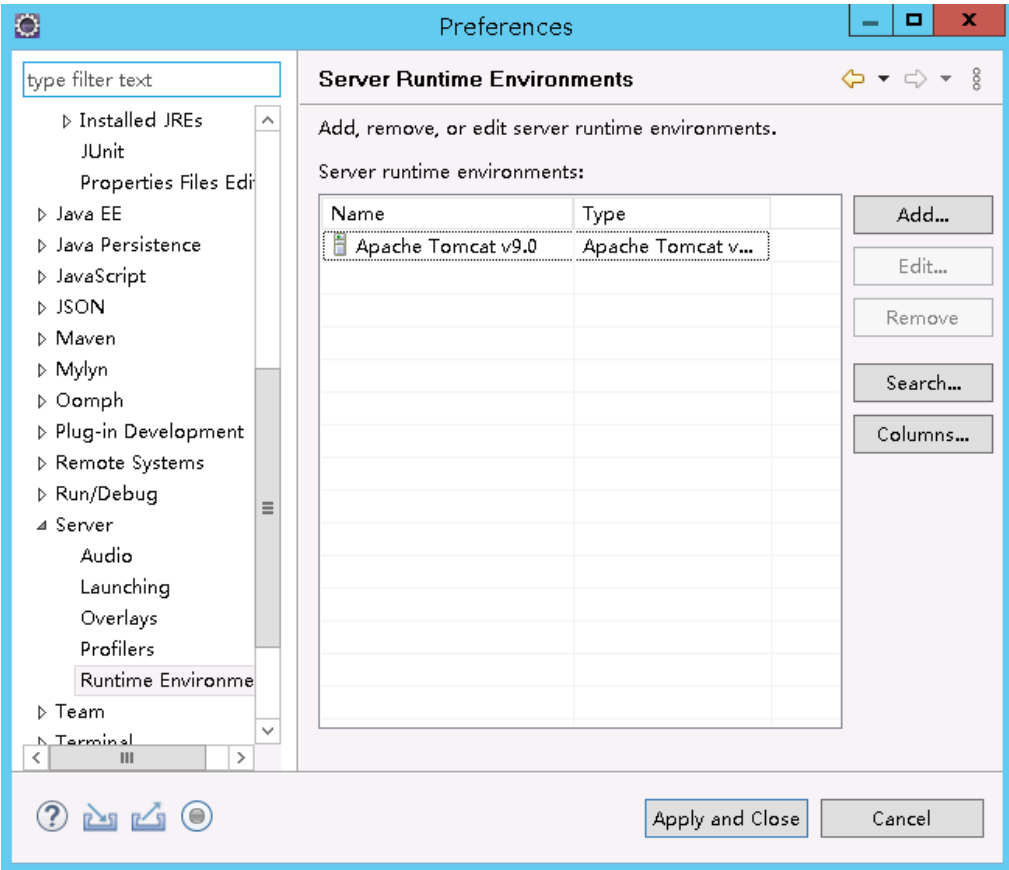


图 5.4 eclipse 中配置 Tomcat 环境

！注：为了响应微信公众号配置 URL 只能为 80 端口和 443 端口的请求，需要在其配置文件 server.xml 中将配置信息：

```
<Connector port="8080" protocol="HTTP/1.1"
            connectionTimeout="20000"
            redirectPort="8443" />
```

中 port="8080"改成 port="80"。

！注：并且 Java EE 下配置 Tomcat 路径应在上述配置更改以后在进行。否则会出现现在 eclipse 下启动的 Tomcat 对应的工程仍然只支持 8080 端口而不支持 80 端口。

5.2 微信公众号注册与选择

微信公众号的类型有订阅号、服务号、企业号。根据注册时所选类型也拥有不同的权限去调用各种接口，以及调用各个接口的次数也会根据所选类型有不同的限制。三种不同的公众号类型的不同区别如图 5.5 所示：



图 5.5 三种微信公众号的区别

订阅号适用于个人或者小型团体的开发，仅作为一个类似于新闻类推广的平台来使用，大部分服务类型接口，如微信支付等是没有权限调用的，因此为支持企业开发应该选择使用服务号或者企业号。在本设计中采用服务号来进行设计。

5.3 开发者模式配置说明

在网页端平台登录后是一个编辑模式的界面，在里面可以自己选择各种功能直接进行编辑，相当于一个微信后台的界面，适用于非开发人员进行操作的后台管理界面。当开发者需要有自身的一个后台平台来管理实现公众号的功能，则需要启用开发者模式。

开发者模式需要在基本配置一栏里对服务器配置进行填写，如图 5.6（a）所在位置。

点击修改配置会进入配置的页面，如图 5.6(b)所示。这里要对四个信息进行说明。



(a)



(b)

图 5.6 开发者模式启用的位置与配置信息

5.3.1 URL

URL 即是开发本地的地址，根据本地主机是个人主机还是云服务器有不同的情况。

对于个人主机来说，本地的 ip 地址在公网上是不能正常访问的，而填写的 URL 的要求之一就是填写的服务器地址能被公网所访问，因此需要经过映射工具将本地的私网地址映射到公网地址上，像花生壳，ngrok 等映射工具就可以实现这一操作。

而云服务器自身的 ip 就是从公网上分配的，能在公网上直接访问的地址，所以此处直接填写 ip 地址是没有问题的，但一般情况下这么填写并不安全，所以不建议直接填写 ip 地址作为 URL。一般我们选择注册一个域名，并在备案之后通过 DNS 解析服务解析到云服务器的 ip 地址，这样我们可以通过域名进行访问，既方便又较为安全，特殊情况下还可以申请 SSL 证书来通过 https 开头的域名来进行访问，提升安

全性。

此外 URL 所填写的地址所创建的 web 项目的 web.xml 文件里须说明连接时指向的 servlet 和 url-pattern 作为填写 url 的准备，即 URL 的正确书写格式应为 http(s)://域名/工程名/url-pattern。而且此 URL 仅支持 80 端口和 443 端口，务必在外网映射的时候选择正确端口，或者云服务器需要在控制台安全组件配置规则中添加 80 端口和 443 端口的入方向规则，允许外网通过这两个端口访问到本地的 Web 工程。

5.3.2 Token

Token 由开发者自己填写，是用于生成签名的。当提交内容时，微信后台会将填写的 Token 与 URL 中包含的 Token 对比，验证安全性。

5.3.3 EncodingAESKey 和消息加解密方式

这两项是用于加密保护，提升安全性的内容。因开发要求，我们暂时先采取随机生成 EncodingAESKey 并且消息加解密方式设为明文模式，安全性较低，为了开发方便而使用。

5.4 开发者模式接入

开发者模式接入需要在上述配置界面填写相应内容的同时，本地的项目也需要进行相应代码的接应，即要实现验证从微信服务器接收的消息的准确性的功能。

首先，配置信息提交后，微信后台服务器会发送 GET 请求到本地服务器上，并且携带四个变量参数：signature、timestamp（时间戳）、nonce（随机数）、echostr（随机字符串）。

其中 signature 是微信加密签名，它包含开发者填写的 token 参数和请求中的 timestamp 参数、nonce 参数，是用于验证签名的重要参数。加密/校验流程如下：

- 1) 将 token、timestamp、nonce 三个参数进行字典序排序；
- 2) 将三个参数字符串拼接成一个字符串进行 sha1 加密；
- 3) 开发者获得加密后的字符串可与 signature 对比，标识该请求来源于微信。^[12]

一旦确定该请求来源于微信，服务器会将参数 echostr 原样发送回微信后台服务器，若接入生效，至此过程开发者模式接入成功。如图 5.7 所示为确认签名的函数。

```

private static final String token = "wechat";
public static boolean checkSignature(String signature,String timestamp,String nonce){
    String[] arr = new String[]{token,timestamp,nonce};
    //排序
    Arrays.sort(arr);

    //生成字符串
    StringBuffer content = new StringBuffer();
    for(int i=0;i<arr.length;i++){
        content.append(arr[i]);
    }

    //sha1加密
    String temp = getSha1(content.toString());

    return temp.equals(signature);
}

```

图 5.7 校验 signature 信息的函数

在此过程中需要注意的是微信服务器所发的是 GET 请求，因此本地的 servlet 需要创建一个类用于处理这个请求，并且要继承一般用于处理网络传输服务的 `HttpServlet` 类，并且调用它的 `doGet` 方法来处理发送来的 GET 请求和其数据流。要调用这个方法需要导入 `httpclient.jar` 和 `httpcore.jar` 两个工具包，并且实行 `doGet` 方法需要穿两个参数，一个 `HttpServletRequest` 对象和一个 `HttpServletResponse` 对象，从 `HttpServletRequest` 对象中获取微信服务器传来的四个参数，之后通过编写函数实现上述加密/校验流程，将经过加工处理后的检验结果进行核对，当核对成功后，通过 `HttpServletResponse` 对象将 `echostr` 传回微信服务器。其函数书写格式为

```

protected void doGet(HttpServletRequest req, HttpServletResponse resp)throws
    ServletException, IOException

```

因为网络传输等因素，可能会有无法接受到 GET 请求或者服务异常的情况，需要添加 `throws` 语句表明该函数可能会抛出异常的情况。

5.5 access_token 的获取

`access_token` 是公众号的全局唯一接口调用凭据，公众号调用各接口时都需使用 `access_token`，像是图片等素材的上传还有自定义菜单的创建都需要事先获取 `access_token`。

`Access_token` 调用请求方式是 GET `https://api.weixin.qq.com/cgi-bin/token?grant_type=client_credential&appid=APPID&secret=APPSECRET`，其中 `AppID` 和 `AppSecret` 可在“微信公众平台-开发-基本配置”页中获得。调用该接口时，需要提前将服务器 IP 地址添加到 IP 白名单中，否则本地服务器无法正常调用。服务器通过 `doGet` 方法

借用 `HttpGet` 类从该地址上获取 JSON 数据包并通过 `execute()`和 `getEntity()`来处理，数据包格式如下：

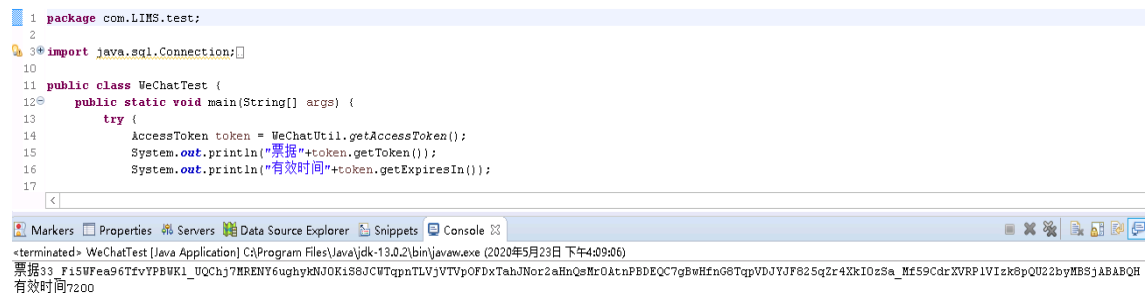
```
{"access_token":"ACCESS_TOKEN","expires_in":7200}
```

获取数据包后因在本地服务器中建立一个类用于储存当前的 `access_token`,方便各种接口调用。如图 5.8 所示为获取 `access_token` 的函数。

```
/**
 * 获取accessToken
 * @return
 * @throws ParseException
 * @throws IOException
 */
public static AccessToken getAccessToken() throws ParseException, IOException {
    AccessToken token = new AccessToken();
    String url = ACCESS_TOKEN_URL.replace("APPID", APPID).replace("APPSECRET", APPSECRET);
    JSONObject jsonObject = doGetStr(url);
    if (jsonObject != null) {
        token.setToken(jsonObject.getString("access_token"));
        token.setExpiresIn(jsonObject.getInt("expires_in"));
    }
    return token;
}
```

图 5.8 获取 `access_token` 的函数

如图 5.9 所示为通过在 Tomcat server 上执行 Java Application 的方式获取 `access_token` 的结果。



```
1 package com.LIMS.test;
2
3 import java.sql.Connection;
4
5
6
7
8
9
10
11 public class WeChatTest {
12     public static void main(String[] args) {
13         try {
14             AccessToken token = WeChatUtil.getAccessToken();
15             System.out.println("票据"+token.getToken());
16             System.out.println("有效时间"+token.getExpiresIn());
17         }
18     }
19 }
```

Console Output:

```
<terminated> WeChatTest [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe (2020年5月23日 下午4:09:06)
票据33_Fi5WFea96TivYPBVRK1_UQChj7HRENY6ughyKJOKiS8JCUTqpnTLVjVTvpOFDxTahjNor2aHnQsMr0atnPBDEQC7gBwHfnG8TqpVDJYJF82Sg2r4XkIOcSa_Mf59CdrXVRP1VIZk8pQU22byMBSjABABQH
有效时间7200
```

图 5.9 `access_token` 的获取

途中票据后面就是本地服务器获得的 `access_token` 值，有效时间的单位为 s。

关于 `access_token` 还有一点说明：

`access_token` 的有效期目前为 2 个小时，需定时刷新，重复获取将导致上次获取的 `access_token` 失效。在本地服务器上一般都是选择要调用 API 时同时获取 `access_token`，尽管在有效时间内也照样更新。

5.6 自定义菜单的创建

自定义菜单是微信公众号的一大特色，可以更方便用户通过菜单寻找功能。

而自定义菜单并不是一个实时事件，它与微信接收用户发送的消息不同，它通过

服务器后台直接提交一个自定义菜单的类型到微信后台提供的地址上，一旦生成成功，这个菜单就是持久的，能够长久性使用的，除非删除它，或者重新提交一个新的菜单去覆盖它。

如图 5.10 所示为菜单创建过程的流程图。

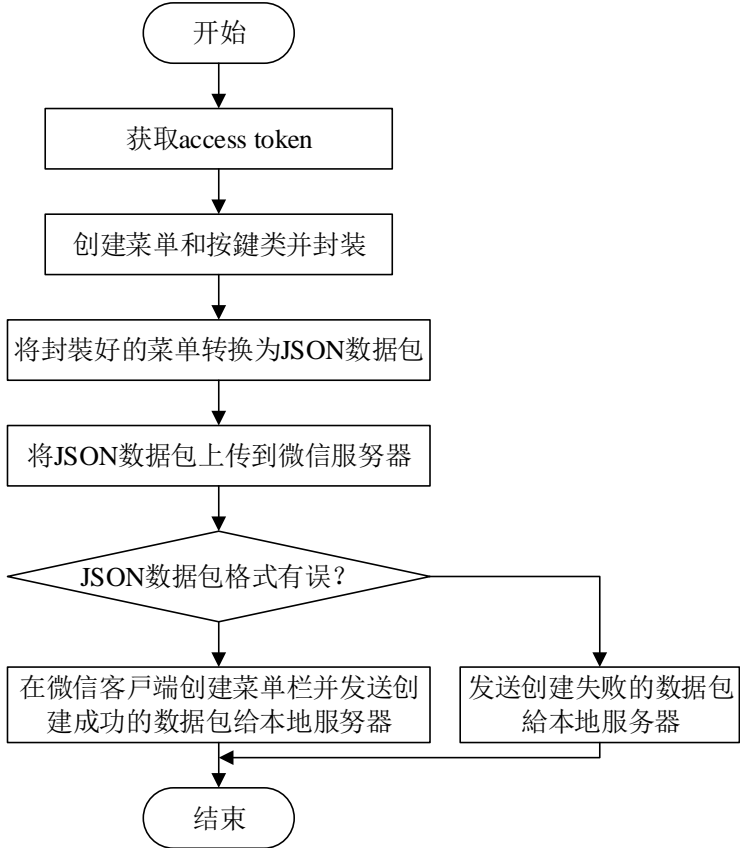


图 5.10 自定义菜单创建过程

创建接口的调用请求为 POST https://api.weixin.qq.com/cgi-bin/menu/create?access_token=ACCESS_TOKEN，将其中的“ACCESS_TOKEN”替换为上文获取到的 access_token。采用 doPost 方法需要用到 HttpPost 类来进行请求，并将已经事先编写好的自定义菜单经过封装，转化为 UTF-8 的形式传输到上述地址，会接收到改地址返回的一个 JSON 数据包：

```
{"errcode":0,"errmsg":"ok"}
```

当 errcode 值为 0 时，说明提交的自定义菜单创建成功，此时便能够在微信公众号上看到菜单。如图 5.11 所示为实现创建菜单的函数，它内部实现了将封装好的菜单转化为 JSON 包并上传至微信服务器上，并获取接收到 errcode，而图 5.12 为通过在 Tomcat server 上执行 Java Application 的方式上传自定义菜单的结果。

```
public static int createMenu(String token,String menu) throws ParseException, IOException{
    int result = 0;
    String url = CREATE_MENU_URL.replace("ACCESS_TOKEN", token);
    JSONObject jsonObject = doPostStr(url, menu);
    if(jsonObject != null){
        result = jsonObject.getInt("errcode");
    }
    return result;
}
```

图 5.11 创建菜单函数

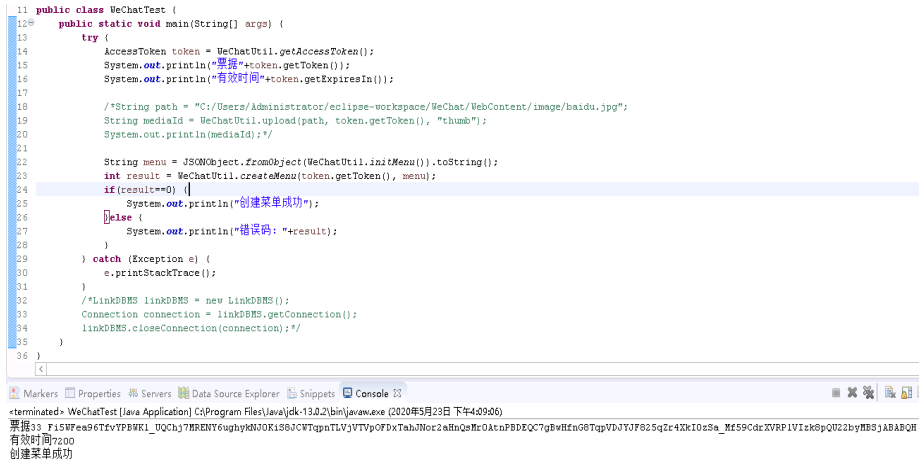


图 5.12 自定义菜单创建成功

如图 5.13 为公众号上自定义菜单创建后的效果图：



图 5.13 微信公众号界面的自定义菜单

当然，返回值 errcode 有可能会有不为 0 的情况，当返回其它错误码时，需到微信开放文档查明该错误码的用意。

一般来讲自定义菜单提交失败时，不考虑网络和 access_token 过期等原因，最有可能的原因就是菜单类型的格式类型出错。

首先，菜单有一级菜单，二级菜单，图 5.13 中最底部的三个菜单就是一级菜单，而点击它们衍生出来的菜单被称为二级菜单，当然不一定需要二级菜单，可以直接给一级菜单的按钮设置类型使用。

菜单的按钮类型有很多，考虑到实际需要只介绍 click 和 view 型按钮的作用。

点击推事件用户点击 click 类型按钮后，微信服务器会通过消息接口推送消息类型为 event 的结构给开发者，并且带上按钮中开发者填写的 key 值，开发者可以通过

自定义的 key 值与用户进行交互^[13]。例如简介为 click 类型时，点击简直，后台服务器根据收到的 key 值决定自动回复一份简介出来。

而跳转 URL 用户点击 view 类型按钮后，微信客户端将会打开开发者在按钮中填写的网页 URL，可与网页授权获取用户基本信息接口结合，获得用户基本信息^[13]。例如查询功能需要跳转到其它页面进行查询，这时候就需要将其按钮类型设为 view 并更改其变量 url。

！注意：当因为功能需求而需要改变某个按钮类型时应注意每个类型按钮的属性都各不一样，像 view 按钮没有 key 值，而 click 按钮没有 url 值。

5.7 消息和事件接收

5.7.1 普通消息接收

当普通微信用户向公众账号发消息时，微信服务器将 POST 消息的 XML 数据包发送到开发者填写的 URL 上。同样的，本地服务器在处理数据包后也要将回复的消息封装成 XML 数据包返还到微信服务器上。其数据交互原理如图 5.14 所示。

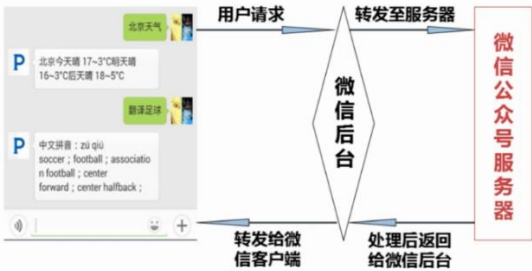


图 5.14 数据交互原理

因为微信处理消息使用的是 XML 格式的特殊性，需要先将 XML 数据包转化为能够正常处理的数据流。为此导入一个工具包 dom4j-1.6.1.jar，它内部包含能够处理 xml 数据包的类 SAXReader，并通过 Document 类储存解析后的数据，最后通过 Element 获取其中的根元素。方法如图 5.15 所示。

```
public static Map<String, String> xmlToMap(HttpServletRequest request) throws IOException, DocumentException {
    Map<String, String> map = new HashMap<String, String>();
    SAXReader reader = new SAXReader();

    InputStream ins = request.getInputStream();
    Document doc = reader.read(ins);

    Element root = doc.getRootElement();

    List<Element> list = root.elements();

    for (Element e : list) {
        map.put(e.getName(), e.getText());
    }
    ins.close();
    return map;
}
```

图 5.15 XML 数据包转化为 map 数据流

而接下来要处理 map 数据流中的元素，图 5.16 为 XML 数据包的组成：

```
<xml>
  <ToUserName><![CDATA[toUser]]></ToUserName>
  <FromUserName><![CDATA[fromUser]]></FromUserName>
  <CreateTime>1348831860</CreateTime>
  <MsgType><![CDATA[text]]></MsgType>
  <Content><![CDATA[this is a test]]></Content>
  <MsgId>1234567890123456</MsgId>
</xml>
```

图 5.16 文本消息 XML 数据包

其中 ToUserName 为开发者微信号，FromUserName 为发送方账号，在消息回复时会将两参数互换。CreateTime 是消息创建时间，在消息返还时通过函数 Date().getTime() 设为当前时间。MsgType 为消息类型，一般用户发送消息均视为文本消息。Content 为发送的文本消息的内容，根据 Content 的内容，进行条件判断，决定回复的消息。

5.7.2 接收事件推送

根据需求，本工程只研究两种较为重要的事件推送机制：

(1) 关注事件

用户在关注公众号后，微信会把这个事件推送到开发者填写的 URL。如图 5.17 为该事件的 XML 数据包。

```
<xml>
  <ToUserName><![CDATA[toUser]]></ToUserName>
  <FromUserName><![CDATA[fromUser]]></FromUserName>
  <CreateTime>123456789</CreateTime>
  <MsgType><![CDATA[event]]></MsgType>
  <Event><![CDATA[subscribe]]></Event>
</xml>
```

图 5.17 关注事件 XML 数据包

其中 ToUserName, FromUserName, CreateTime 同文本消息，MsgType 的值为 event，说明该数据包是事件推送，而 Event 为事件类型，subscribe 代表关注事件。

(2) 点击菜单拉取消息时的事件推送

当用户点击的菜单按钮为 click 类型时，会发送此事件推送到开发者填写的 URL 上。注意，点击一级菜单只会弹出子菜单，不会上报到本地服务器上。如图 5.18 为该事件的 XML 数据包。


```
<xml>
  <ToUserName><![CDATA[toUser]]></ToUserName>
  <FromUserName><![CDATA[FromUser]]></FromUserName>
  <CreateTime>123456789</CreateTime>
  <MsgType><![CDATA[event]]></MsgType>
  <Event><![CDATA[CLICK]]></Event>
  <EventKey><![CDATA[EVENTKEY]]></EventKey>
</xml>
```

图 5.18 菜单事件 XML 数据包

其中 ToUserName, FromUserName, CreateTime, MsgType 同关注事件, Event 值为 CLICK, 代表是 click 按钮引起的事件。EventKey 是事件 KEY 值, 与自定义菜单接口中 KEY 值对应。

5.8 消息和事件的自动回复功能

如图 5.19 所示为这整个自动回复过程的流程图。

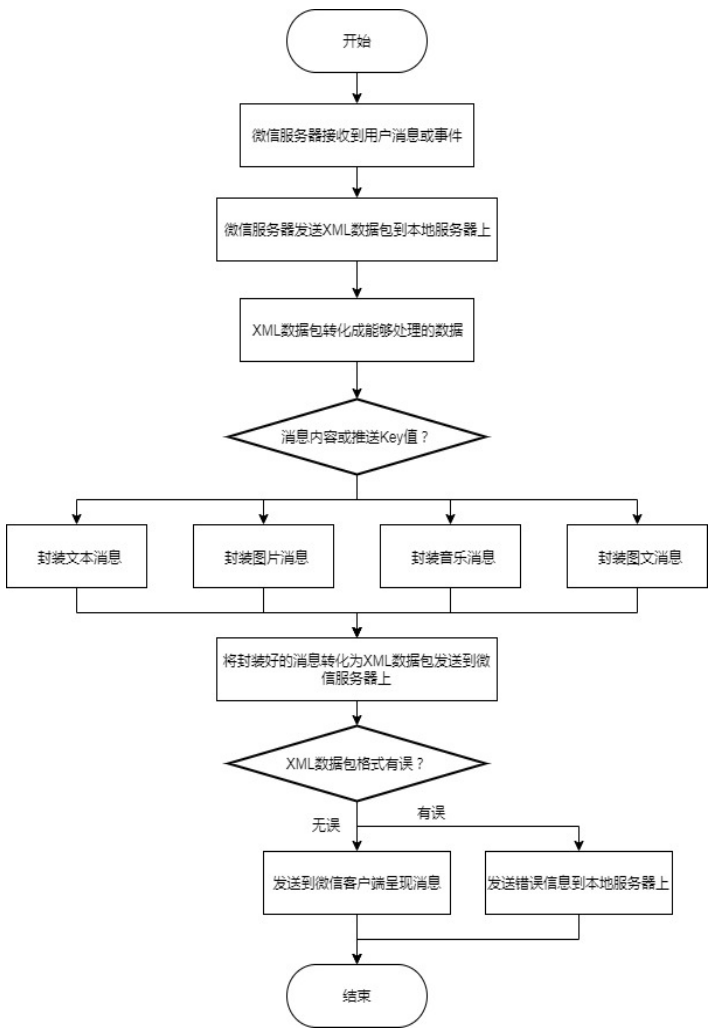


图 5.19 自动消息回复的流程图

根据前面提到接收到的消息的 Content 或者事件的类型, KEY 值等来根据需要回复相应的内容。与接收到的不同类型消息的 XML 数据包一样, 回复的消息也可以根据需要选择不同类型的消息, 最后封装成 XML 数据包返还给微信服务器。因为接收消息的接口通过 POST 请求实现, 因此数据包的回复也同样在 doPost 方法中实现, 本身并不作为一个接口使用。

在处理收到的 XML 数据包时使用的是 SAXReader, 在将组装好的消息转化为 XML 数据包所使用的的是 Xstream。这是一个 Java 类库, 用来将对象序列化成 XML (JSON) 或反序列化为对象。

回复的消息类型大致分为文本消息, 图片消息, 音乐消息和图文消息。其中文本消息的 XML 数据包格式与图 5.16 中的格式几乎一致, 只需注意将 ToUserName 和 FromUserName 的内容互换。如图 5.20 为回复文本消息的效果图。



图 5.20 回复文本消息

而后面三种消息的格式以及处理都较为复杂, 下面将一一指出:

5.8.1 图片消息的回复

如图 5.21 为图片消息的 XML 数据包格式:

```
<xml>
  <ToUserName><![CDATA[toUser]]></ToUserName>
  <FromUserName><![CDATA[fromUser]]></FromUserName>
  <CreateTime>12345678</CreateTime>
  <MsgType><![CDATA[image]]></MsgType>
  <Image>
    <MediaId><![CDATA[media_id]]></MediaId>
  </Image>
</xml>
```

图 5.21 图片消息 XML 数据包

其中需要注意的是, MsgType 为固定值 image, 而 MediaId 是通过接口上传该文件从微信服务器获得的 id。即为了长期使用这些多媒体文件, 如图片, 音乐等, 需要先将这些文件上传至微信服务器方便使用。

素材上传的接口需要通过 GET 请求来实现, 其中需要先打包好输入流, 根据提供的 access_token 和相应类型 TYPE 来得到上传的地址, 然后定义 BufferedReader 输

入流来读取 URL 的响应，获取 JSON 数据包，最后通过 JSONObject 类获取其中的 meida_id 来使用。

如图 5.22 为上传成功一张照片后的结果：



图 5.22 素材上传成功返回值显示

其中最后一行的字符串就是我们所获得的上传至微信服务器中图片的 media_id，我们可以直接通过赋值调用微信服务器中的这张图片，如图 5.23 为效果图：



图 5.23 回复图片消息

5.8.2 音乐消息的回复

如图 5.24 为音乐消息的 XML 数据包格式，MsgType 为固定值 music，需要注意的是 Music 类型中的属性，ThumbMediaId 同图片消息中的 MeidaId 一样为上传至微信服务器的素材 id，Title 为标题，Description 为描述，MusicURL 和 HQMusicUrl 均为音乐链接，HQMusicUrl 为高清音乐链接地址，网络环境较好的情况下优先链接到此地址下。

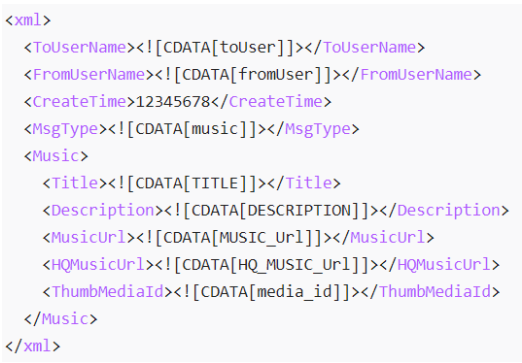


图 5.24 音乐消息 XML 数据包

如图 5.25 为效果图，图 5.26 为跳转链接的音乐播放器界面，当点击图 3.16 回复的消息时回链接到图 5.26 的界面下播放音乐。



图 5.25 回复音乐消息

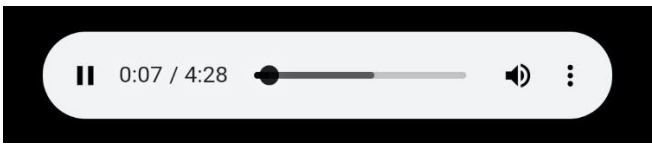


图 5.26 音乐播放界面的音乐播放器

5.8.3 图文消息的回复

如图 5.27 为图文消息的 XML 数据包格式，MsgType 为固定值 news。ArticleCount 为图文消息个数，一般为 1。Title 与 Description 同音乐消息。PicUrl 是图片的链接地址，支持 JPG、PNG 格式，较好的效果为大图 360*200，小图 200*200。Url 为点击图文消息跳转链接。item 代表一个图文消息，而 Articles 为图文消息信息，注意，如果图文数超过限制，则将只发限制内的条数，即 item 可以设多个，但只能显示 ArticleCount 值个数的图文消息。

如图 5.28 为公众号内回复的效果图，图 5.29 为跳转页面，需要点击图 5.28 中的图文消息才能跳转至此页面。

```
<xml>
  <ToUserName><![CDATA[toUser]]></ToUserName>
  <FromUserName><![CDATA[fromUser]]></FromUserName>
  <CreateTime>12345678</CreateTime>
  <MsgType><![CDATA[news]]></MsgType>
  <ArticleCount>1</ArticleCount>
  <Articles>
    <item>
      <Title><![CDATA[title1]]></Title>
      <Description><![CDATA[description1]]></Description>
      <PicUrl><![CDATA[picurl]]></PicUrl>
      <Url><![CDATA[url]]></Url>
    </item>
  </Articles>
</xml>
```

图 5.27 图文消息 XML 数据包



图 5.28 回复图文消息



图 5.29 跳转页面

5.9 微信网页开发

因公众号页面是类似于推广交互界面的类型，业务处理等交互不宜在此界面下进行，不仅是处理业务时交互较为繁琐的原因，还有相较于网页信息传输也不够安全。

在菜单中设计 **view** 类型的按钮，将其 **url** 值设为开发相对应功能的网页 **JSP** 文件上，便可以通过菜单栏进入网页进行业务处理。

这里给出三个查询的设计过程，如图 5.30 所示为其流程图。

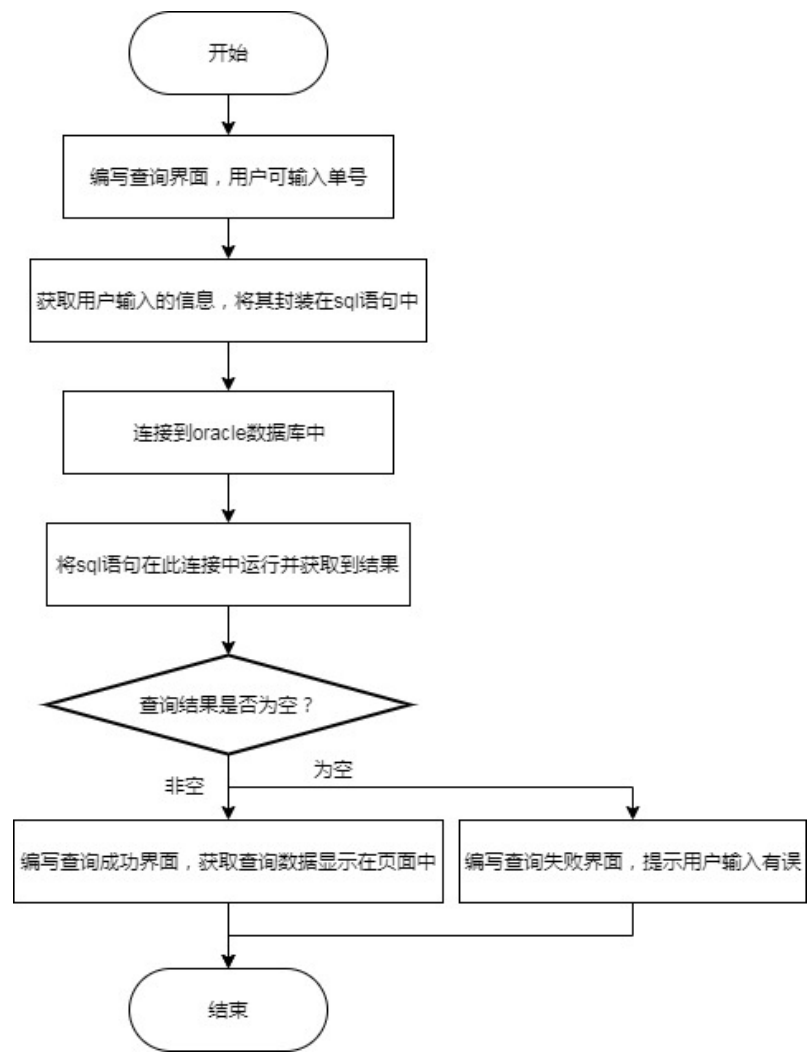


图 5.30 查询过程

Java Web 工程中可以直接创建 JSP 文件，动态生成 HTML 网页，大部分的网页可以通过此方式构建，只需注意在文件开头注明：`<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>`，表明生成的是 HTML 网页，编码为 UTF-8 格式，servlet 编程语言为 java 类型。

项目需求功能有以图文方式显示简介与新闻，通过公众号查询订单进度、证书等，以及在线建议和投诉。我们在工程中需要将每一个功能需要的 JSP 页面设计完毕，并通过 POST 方法与后台 java 文件进行交互。

在设计页面这一块微信有一套网页开发样式库 WeUI，

该样式库目前包含表单、基础组件、操作反馈、导航相关、搜索相关、层级规范等内容。在 JSP 文件中添加标签`<link rel="stylesheet" href="https://res.wx.qq.com/open/libs/weui/2.3.0/weui.min.css"/>`，即可在主体的标签中根据 class 值获取到需要的样式，并传送到后台进行处理。如图 5.31 为 WeUI 的各组件预览界面：



图 5.31 WeUI 的预览界面

因为简介新闻功能只需要简单的前端网页设计就能实现，因此这里重点说明查询功能的实现过程。

5.9.1 查询进度

先要明白表单标签的使用，为了方便处理我们需要在 JAVA 文件中处理从 JSP 页面下传来的用户填写的参数来实现功能，因此要用到<form>/<input>标签。<form>标签就是最基础的表单标签，在它内部的定义 input 标签是实现用户在此界面的输入。在<form>标签中需要添加属性 action="/WeChat/processQuery" method="post"，其中 action 所指向处理 form 标签中用户输入数据的 JAVA 文件，在该 JAVA 文件中应注明 @WebServlet("/processQuery")，以说明该页面下提交信息后信息将会传输到该 JAVA 文件下进行处理。而 method 方法可以有 get 和 post 两种，选择 post 请求在处理完数据后，还会将处理好的数据发送至另一个页面并使网页跳转至另一个页面下显示结果。如图 5.32 为该页面的 html 代码：

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width,initial-scale=1.0">
<link rel="stylesheet" href="https://res.wx.qq.com/open/libs/weui/2.3.0/weui.min.css"/>
<title>进度查询</title>
</head>
<body>
<div style="height:100px"></div>
<div class="page_hd">
<h1 class="page_title" align="center">进度查询</h1>
</div>
<form action="/WeChat/processQuery" method="post">
<div class="weui-form_control-area">
<div class="weui-cell weui-cell_active">
<div class="weui-cell_hd"><label class="weui-label">委托单号</label></div>
<div class="weui-cell_bd">
<input id="js_input" type="text" name="ordernumber" class="weui-input" placeholder="填写委托单号">
</div>
</div>
</div>
<div class="weui-form_opr-area">
<input class="weui-btn weui-btn_primary weui-btn_disabled" type="submit" value="确定">
</div>
</form>
</body>
</html>
```

图 5.32 查询进度页面的代码实现

之后将查询的 sql 语句封装成 PreparedStatement 类成员，并执行 sql 语句获得结果，该 sql 语句为“select EQUIPNAME, NORMTYPE, FACTORYNO, EQUIPSTATUS, CERTSTATUS from Detect where CONNO=?”，其中? 为从 JSP 页面获取到的单号，而 Detect 表所包含的属性：委托单号(Conno)，仪器名称(equipname)，规格型号(normtype)，出厂编号(factoryno)，仪器状态(equipstatus)，证书状态(certstatus)，其查询代码如图 5.33 所示。

```
LinkDBMS linkDBMS = new LinkDBMS();
Connection conn = linkDBMS.getConnection2();
String sql="select EQUIPNAME, NORMTYPE, FACTORYNO, EQUIPSTATUS, CERTSTATUS from Detect where CONNO=?";
try {
    PreparedStatement ps = conn.prepareStatement(sql);
    ps.setInt(1, conno);
    ResultSet rs = ps.executeQuery();
    while(rs.next()) {
        equipname = rs.getString("EQUIPNAME");
        normtype = rs.getString("NORMTYPE");
        factoryno = rs.getString("FACTORYNO");
        equipstatus = rs.getInt("EQUIPSTATUS");
        certstatus = rs.getInt("CERTSTATUS");
    }
    rs.close();
    ps.close();
    conn.close();
    if("".equals(equipname)){
        req.getRequestDispatcher("/ErrorPage.jsp").forward(req, resp);
    }
    else {
        req.setAttribute("ordernumber", ordernumber);
        req.setAttribute("equipname", equipname);
        req.setAttribute("normtype", normtype);
        req.setAttribute("factoryno", factoryno);
        req.setAttribute("equipstatus", equipstatus);
        req.setAttribute("certstatus", certstatus);
        req.getRequestDispatcher("/result1.jsp").forward(req, resp);
    }
} catch (SQLException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
```

图 5.33 查询代码

5.9.2 查询证书状态

同样通过 POST 请求获取 cert.jsp 中填写的证书编号，<form>中的标签属性为 action="/WeChat/certQuery" method="post"，CertQuery.java 文件中应注明 @WebServlet("/certQuery")。

在页面上输入证书编号并点击确定后，certQuery 会以 POST 请求获得证书编号，并通过输入证书编号，显示证书单位(customername)，仪器名称(equipname)，规格型号(normtype)，出厂编号(factoryno)，检定日期(detectdate)，这些元素分散在 Detect、Cert 和 Contract 表中，因此 sql 语句为：“select CUSNAMECERT, EQUIPNAME, NORMTYPE, FACTORYNO, DETECTDATE from CERT,DETECT,CONTRACT where CERT.CERTNO=? andCERT.DETECTID=DETECT.DETECTID and DETECT.CONNO=CONTRACT.CONNO;”涉及到多表查询和属性自然连接时合并的问题，其中? 为证书编号。代码部分与查询进度基本一致。

5.9.3 查询仪器

同样通过 POST 请求获取 equip.jsp 中填写的证书编号，<form>中的标签属性为 action="/WeChat/itemQuery" method="post"，ItemQuery.java 文件中应注明 @WebServlet("/itemQuery")。

再页面上输入证书编号并点击确定后，itemQuery 会以 POST 请求获得仪器名称，并通过输入仪器名称，显示仪器名称（itemname），测量范围（rangeformat），不确定度（uncertaintyformat），sql 语句为：“select RANGEFORMAT, UNCERTAINTYFORMAT from V_CNASITEM where ITEMNAME=?” 其中？为仪器名称，查询时自动替换。代码部分与查询进度基本一致。

！注：仪器查询输入的是中文，和查询证书与进度时不同，因为 UTF-8 编码的格式为"8859_1"，需要先将将得到的这一字符串进行编码格式的转换。有多种方式，该项目选择在 req.getParameter("itemname")前注明代码 req.setCharacterEncoding("GB2312");（如图 5.34 所示）。

```
protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {  
    req.setCharacterEncoding("GB2312");  
    String itemname = req.getParameter("itemname");
```

图 5.34 编码格式的转换

5.9.4 在线意见

查询功能类似，通过 POST 方法获取 JSP 页面中标签 TEXTAREA 中用户所填写的建议，以及允许用户上传图片，通过 uploader 组件来实现，并且在 Javascript 中编写上传文件的函数。

在实现的页面上点击确定以后，在 suggestionReceiver.java 文件中直接将获取到的文本域中的意见通过 system.out.println()函数输出到控制台，方便后台看到用户的意见，并且进一步将获取到的上传的图片通过 DiskFileItemFactory、ServletFileUpload 和 FILE 类的处理保存到本地的存放图片的路径下。其主要代码实现如图 5.35 所示，控制台输出如图 5.36 所示。

```
public class suggestionReceiver extends HttpServlet{
    protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
        final String upload_direct= "C:/Users/Administrator/eclipse-workspace/WeChat/WebContent/image";
        final int MEMORY_THRESHOLD = 1024 * 1024 * 3; // 3MB
        final int MAX_FILE_SIZE = 1024 * 1024 * 40; // 40MB
        final int MAX_REQUEST_SIZE = 1024 * 1024 * 50; // 50MB
        DiskFileItemFactory factory = new DiskFileItemFactory();
        factory.setSizeThreshold(MEMORY_THRESHOLD);
        factory.setRepository(new File(System.getProperty("java.io.tmpdir")));
        ServletFileUpload upload = new ServletFileUpload(factory);
        upload.setFileSizeMax(MAX_FILE_SIZE);
        upload.setSizeMax(MAX_REQUEST_SIZE);
        upload.setHeaderEncoding("UTF-8");
        String uploadPath = getServletContext().getRealPath("/") + File.separator + upload_direct;
        try {
            // 解析请求的内容提取文件数据
            @SuppressWarnings("unchecked")
            List<FileItem> formItems = upload.parseRequest((RequestContext) req);

            if (formItems != null && formItems.size() > 0) {
                // 迭代表单数据
                for (FileItem item : formItems) {
                    // 处理不在表单中的字段
                    if (!item.isFormField()) {
                        String fileName = new File(item.getName()).getName();
                        String filePath = uploadPath + File.separator + fileName;
                        File storeFile = new File(filePath);
                        // 在控制台输出文件的上传路径
                        System.out.println(filePath);
                        // 保存文件到硬盘
                        item.write(storeFile);
                    }
                }
            }
        } catch (Exception ex) {
            req.setAttribute("message",
                "错误信息: " + ex.getMessage());
        }
    }
}
```

图 5.35 读取上传文件信息并保存在本地路径中

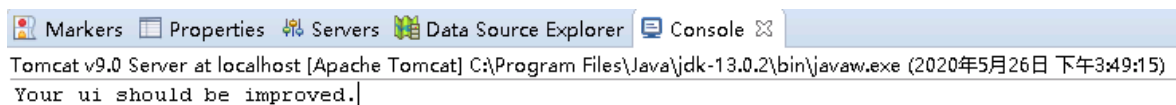


图 5.36 控制台输出用户意见

5.10 微信网页授权

如果用户在微信客户端中访问第三方网页，公众号可以通过微信网页授权机制，来获取用户基本信息。如果用户同意授权，公众号可以获取到网页授权 access_token，通过网页授权 access_token 可以获取用户基本信息等等。

此处的 access_token 与调用微信公众号 API 接口的 access_token 凭证不同。另一方面，在获取授权之前，需要在接口权限配置项中，在网页授权一栏添加回调域名。在微信公众号设置中，此配置项只能填写域名，且必须是经过工信部备案的域名，工信部备案的时间一般在 3-20 天，因此我们选择微信公众平台测试账号来实现其功能。在测试号中，此配置项不限于域名的填写，能够通过公网访问的 ip 地址也同样可以被支持，如图 5.37 所示。填写完以后可以着手微信授权页面的开发。

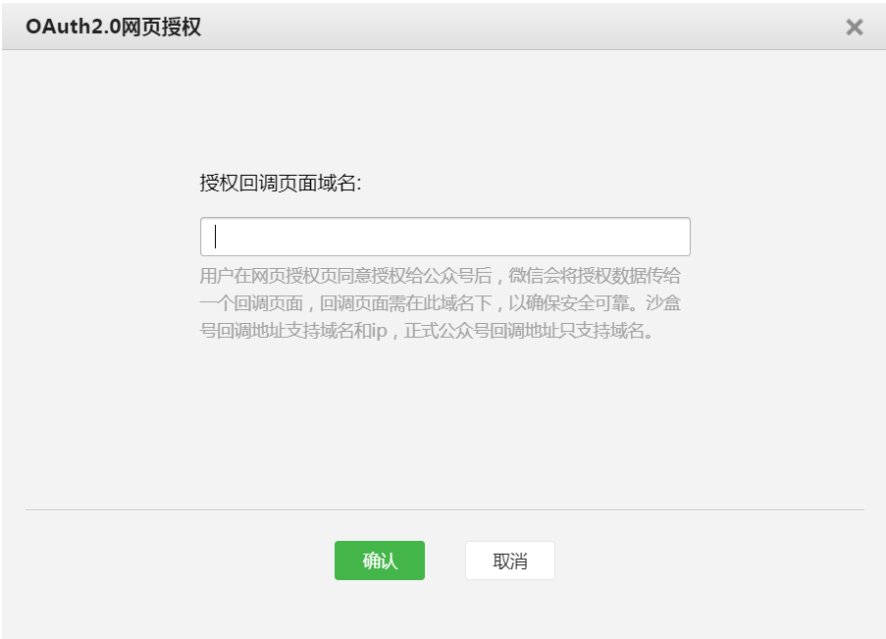


图 5.37 测试号回调地址要求

具体而言，网页授权流程分为四步^[14]：

- 1、引导用户进入授权页面同意授权，获取 code
 - 2、通过 code 换取网页授权 access_token（与基础支持中的 access_token 不同）
 - 3、如果需要，开发者可以刷新网页授权 access_token，避免过期
 - 4、通过网页授权 access_token 和 openid 获取用户基本信息（支持 UnionID 机制）
- 其中第 3 步并非次工程必要步骤，不予以讨论。

5.10.1 用户同意授权，获取 code

如图 5.38 所示为此步骤实现的函数方法，先引导用户打开图中更改部分参数的 url 地址，scope 是较为特殊的参数。一般情况下 scope 参数默认填写为 snsapi_userinfo，图 5.39(a)和(b)为初次登陆和再次登陆时 scope 等于 snsapi_userinfo 时的授权页面。

```
@WebServlet("/wxLogin")
public class LoginServlet extends HttpServlet{
    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
        String backUrl = "https://open.weixin.qq.com/connect/oauth2/authorize?appid="+ AuthUtil.APPID
            + "&redirect_uri=" + URLEncoder.encode(backUrl)
            + "&response_type=code"
            + "&scope=snsapi_userinfo"
            + "&state=STATE#wechat_redirect";
        resp.sendRedirect(url);
    }
}
```

图 5.38 用户打开授权页面的函数



(a)



(b)

图 5.39 scope 等于 snsapi_userinfo 时的授权页面

！注：测试号尽管在 scope 等于 snsapi_userinfo 的情况下也不会出现如图 5.39(a) 的界面，而是直接跳转至回调域名，在正式公众号才会正常显示。

如果用户同意授权，页面将跳转至 redirect_uri/?code=CODE&state=STATE。

5.10.2 通过 code 换取网页授权 access_token

如图 5.40 所示为此步骤实现的函数方法，其中申请的链接中同样需要进行参数的替换，这里的 code 取上一步获得的值，grant_type 取固定值 authorization_code。

```
protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
    String code = req.getParameter("code");
    String url = "https://api.weixin.qq.com/sns/oauth2/access_token?appid=" + AuthUtil.APPID
        + "&secret=" + AuthUtil.APPSECRET
        + "&code=" + code
        + "&grant_type=authorization_code";
    JSONObject jsonObject = AuthUtil.doGetJson(url);
    String openid = jsonObject.getString("openid");
    String token = jsonObject.getString("access_token");
}
```

图 5.40 获取网页授权 access_token 的方法

当 code 正确时，该链接会发送一个 JSON 数据包到服务器上，通过 GET 请求获取这个数据，数据如图 5.41 所示。

```
{
  "access_token": "ACCESS_TOKEN",
  "expires_in": 7200,
  "refresh_token": "REFRESH_TOKEN",
  "openid": "OPENID",
  "scope": "SCOPE"
}
```

图 5.41 网页 access_token 等数据

其中除了最重要的 access_token 以外，还要注意传送回来的数据中包括用户的 openid。它是用户唯一标识，在第三方网站上进行微信账号绑定时的凭证。同时当 scope 为 snsapi_base 时只获取用户的 openid，因此到此步就结束了授权。

5.10.3 拉取用户信息(需 scope 为 snsapi_userinfo)

如果网页授权作用域为 snsapi_userinfo，则此时开发者可以通过 access_token 和 openid 拉取用户信息。如图 5.42 所示为获取用户信息的方法，至此微信授权的操作已全部完成。

```
String infoUrl = "https://api.weixin.qq.com/sns/userinfo?access_token=" + token
                + "&openid=" + openid
                + "&lang=zh_CN";
JSONObject userInfo = AuthUtil.doGetJson(infoUrl);
```

图 5.42 获取用户信息的方法

5.11 第三方网站绑定微信账号

在上述过程中经过微信授权获取了用户的 openid，可以在第三方网站上让用户对该网站上的原账号进行绑定。

计量管理平台的数据库所使用的是 Oracle 数据库，在 java 中需要导入 ojdbc12.jar 包，里面包含了连接数据库，查询和更新数据库中数据的函数。

在绑定微信账号的过程中需要访问原计量管理平台数据库中的用户表，并且为了绑定的需要在原用户表中添加了一个属性为 openid，代表用户所绑定的微信账号。需要在工程下先加载数据库驱动，再连接到数据库上，在用户表中查询与该当前登录网页的用户的 openid 相匹配的用户名，若查询到的值不为空，即说明该微信用户已经绑定过本地账号。若查询到的值为空，说明此微信用户还未绑定账号，会链接进绑定

页面进行绑定。在页面中输入用户在计量管理平台登记的用户名和密码，通过 POST 方式获取到后在数据库中的用户表查询，若存在匹配项，则将其 openid 进行更新，即完成了绑定。

！注：查询和更新的过程使用的是 sql 语句，格式的书写应注意，全大写和全小写的情况，而且 sql 语句会在 conn 容器中进行处理，更换其中的未知值，且位置是一一对应的，位置出错同样可能出现查询更新失败的情况。

如图 5.43 为登录绑定界面，图中的用户名密码为手动填写。



图 5.43 登录绑定界面

如图 5.44 为绑定成功的提示页面：



图 5.44 绑定成功界面

再次登录时，如图 5.45 所示的界面：



图 5.45 第二次登录界面

第 6 章 系统测试

工程需求的功能已按照自定义菜单一一给出，如图 5.13 所示，一级菜单栏包括机构，查询与服务。机构的二级菜单中包括简介、新闻、资质、联系方式。查询的二级菜单包括进度、价格与仪器。服务的二级菜单包括在线投诉、建议。

6.1 简介新闻

简介与新闻都是简单的前端页面，几乎没有与后端进行交互的需求，通过文字与图片描述的内容来表现。图 6.1 为部分简介页面：

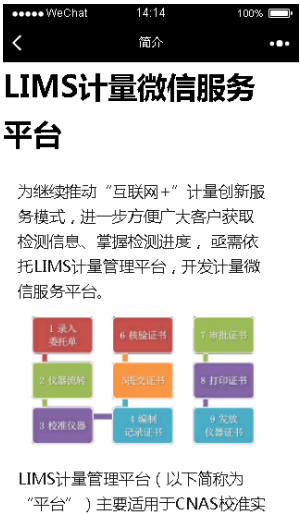


图 6.1 简介界面

其中需要注意的是图片标签，src 属性所填写的值是以“域名+工程名+图片文件名”为格式的，不能填写在此电脑上的绝对路径。图 6.2 为新闻页面：



图 6.2 新闻界面

通过后台 JSP 文件可以在每一个形如图 6.3 的标签中更改图 6.3 中每一个暂无标签的信息来进行新闻的更新。其中第一个新闻为简介，即在图 6.3 中的 a 标签内添加属性 href 指向简介页面的位置，通过点击 LIMS 计量管理平台简介便可进入图 6.2 的界面。这是在后台更新新闻的方式。

```
<a class="weui-cell weui-cell_access">
  <div class="weui-cell_bd">
    <p>暂无</p>
  </div>
  <div class="weui-cell_ft">
    </div>
</a>
```

图 6.3 新闻标签

6.2 查询

三种查询输入以及所要查询的内容都各不相同，因此要用三个不同的页面和处理 HTTP 服务的 JAVA 文件来完成查询功能。

6.2.1 查询进度

图 6.4 为查询进度界面：



图 6.4 查询进度界面

当查询到非空数据时，携带参数跳转页面至查询结果页面，如图 6.5 所示：

×

查询结果

...

查询结果

| | |
|------|-----|
| 委托单号 | |
| 仪器名称 | 千分尺 |
| 规格型号 | 555 |
| 出厂编号 | |
| 仪器状态 | 0 |
| 证书状态 | 0 |

图 6.5 单号查询结果

6.2.2 查询证书状态

图 6.6 为查询证书界面：

×

证书查询

...

证书查询

证书编号

填写证书编号

确定

图 6.6 查询证书界面

当查询到非空数据时，携带参数跳转页面至查询结果页面，如图 6.7 所示：



图 6.7 证书查询结果

6.2.3 仪器检测

图 6.8 为查询仪器界面：



图 6.8 仪器检测界面

当查询到非空数据时，携带参数跳转页面至查询结果页面，如图 6.9 所示：

×

查询结果

...

查询结果

| | |
|------|-------------------|
| 仪器名称 | 通用卡尺 |
| 测量范围 | (0~5000)mm |
| 不确定度 | $U=10\mu\text{m}$ |

图 6.9 仪器检测结果

6.2.4 错误页跳转

为了应对输入格式错误，或输入的内容没有相应数据记录的情况，编写了一个错误页面，当上面三种查询出现错误情况时，会跳转至此界面，界面的图表文字格式参考 WeUI 中的 Icons 组件，如图 6.10 所示：



图 6.10 错误页

6.3 在线意见

如图 6.11 为在线意见的界面，用户可在文本区填写对企业业务的意见，同时可以上传图片。

最后在确定以后会反馈给用户一个成功提交的界面，如图 6.12 所示。

×

建议

...

建议

请将您的意见与问题反馈给我们

问题描述

请描述你所发生的问题

0/200

图片上传

+

确定

◀

○

◻

图 6.11 在线建议界面

×

成功提交

...

✓

意见提交成功

非常感谢您的支持!

◀

○

◻

图 6.12 意见提交成功界面

第 7 章 结论

7.1 总结

微信公众号作为国内一大平台已经根植于人们的生活中，不管是对于个人还是企业，微信公众号都是一个非常值得利用的资源平台去开拓业务，这也是微信公众号不断蓬勃发展的原因。

微信公众号的设计也越来越受关注，因为公众平台本身自带的编辑模式，非开发者能自由利用微信的资源使用公众号。而对于开发者或者企业人员来说，微信公众平台对开发语言的兼容性较高，至今 python 与 php 的微信后台开发较为完整，而网上的 Java 开发文件偏少，但 Java 同样适合开发后台服务，尤其是 JAVA EE 的使用，配合 Tomcat 服务器，即使是个人开发者也能较好上手。

同时 Java 在开发时需要的较多函数可以通过网上下载的各种工具包自带实现，如连接 Oracle 数据库的函数，将 xml 数据流解析的函数，也证明了 Java 开发的优势。

公众号主要通过聊天界面或者网页与用户进行交互，消息的自动回复功能方便而且直观，但是封装 xml 数据包容易出错，一个地方的标签写错都会报错。网页开发可以调用微信的 WeUI 来应用各种较为美观的组件，同时 Java Web 工程支持 JSP 文件与 Servlet 容器之间的交互，大大提高了通过网页与用户交互的能力。

此设计最重要的是实现了页面交互与工程连接数据库交互的结合，实现了两次远程调用的情况，解决了服务器不需要在固定一台主机或云服务器上的情况，数据和工程可以分开放在不同的位置，大大增加了安全性。

同时自定义菜单的随意创建和按钮类型甚至可以允许通过菜单事件链接到小程序，小程序比公众号出现得晚，但已和公众号一样是重要的传播途径，通过公众号能访问小程序，也大大减少了公众号工程中页面交互的页数，减少内存冗余的情况。

但是这些成果都是在测试号下使用的，测试号能够使用所有高级接口，能在测试号下正常运行的工程在服务号下也一定可以。但测试号无必须工信部备案域名的限制，对安全性的考虑较低，且测试号仅开发者本人能够使用，不能通过移动端公众号搜索查询到。因此需要在域名备案成功，配置 SSL 证书后在服务号上运行才能正式被微信用户所使用。

7.2 展望

微信公众号在接下来的岁月里继续发展，成为中国乃至世界一流的自媒体平台。其在开发者模式下功能的开发在近几年已经趋于完善，接下来的发展重点应该在于安全和性能的问题上。

首先开发者模式配置下 URL 支持 https 的域名，证明允许 SSL 证书的使用，提高安全性。还有 JS-SDK 需要备案过的域名作为接口安全域名才可以使用，因为 JS-SDK 提供各种微信的功能接口，如扫一扫，微信支付等，甚至会读取用户信息的情况，必须要有经过认证的企业才能使用这些功能。未来关于这方面的要求会更加严苛，并且也意味着公众号的发展会倾向于服务号、企业号上的研究，订阅号在接下来的一段时间内可能不会有太大的变化。

近些年来因为公众号的发展，许多公司都越来越重视这块，自媒体平台不断增多，公众号接下来会面临的挑战也会越来越大。同时因为它的主体是微信这一社交软件，也证实了它的特殊性成为了一枝独秀。也因此更多的企业会选择微信公众号作为移动端推广的平台。

参考文献

- [1] 开启计量“微时代” 打造“指尖上的服务”——辽宁省计量院开通微信公众服务平台[J], 衡器, 2015, 44(7): 5.
- [2] 李九祥. 企业微信公众号的运营[J], 企业管理, 2019, (4):102-103.
- [3] Xiling Xiong, Xiaofeng Li, Xinrui Wang, Haiting Wu, Lingnan He. Collective Attention in WeChat Public Platform[A], International Conference on Behavioral, Economic, and Socio-Cultural Computing[C], 2018: 261-264.
- [4] 崔雨. 微信公众号的设计研究[D], 合肥:合肥工业大学, 2019.
- [5] W. B. Kuang, Research on the development of WeChat[J], Chinese Journal of Journalism and Communication, 2014, 36(5):147-156.
- [6] Y. Wang, C. P. Deng, et al. Research on Impact of Mobile Social Media on the Knowledge Sharing of Virtual Team[J]. Information Studies:Theory & Application, 2015, 38(11):59-63.
- [7] Feifei Wu, Yiming Tong, Lucheng Huang, Hong Miao, Xin Li. The Application Prospect Analysis of Technology Based on WeChat Official Accounts[A], 2018 Portland International Conference on Management of Engineering and Technology[C],2018: 1-9.
- [8] Xu J, Du Z, Guo J, et al. Empirical and modeling studies of WeChat information dissemination[J]. Physica A: Statistical Mechanics and its Applications, 2018, 512:1113-1120.
- [9] 徐敬宏, 胡世明. 微信公众号信息传播三大规律及其运用[J], 中国出版, 2019(10):3-7.
- [10] 方兴东, 石现升, 张笑容, 张静. 微信传播机制与治理问题研究[J], 现代传播, 2013, (6): 122-127.
- [11] Java（计算机编程语言）_百度百科[EB/OL]. <https://baike.baidu.com/item/Java/85979?fr=aladdin>.
- [12] 接入概述|微信开放文档[EB/OL]. https://developers.weixin.qq.com/doc/offiaccount/Basic_Information/Access_Overview.html.
- [13] 创建接口|微信开放文档[EB/OL]. <https://developers.weixin.qq.com/doc/offiaccount/>

Custom_Menus/Creating_Custom-Defined_Menu.html.

- [14] 网页授权|微信开放文档[EB/OL]. https://developers.weixin.qq.com/doc/offiaccount/OA_Web_Apps/Wechat_webpage_authorization.html.

致 谢

首先衷心感谢高福祥老师的耐心指导，我第一次学习编写 Java Web 工程，有很多不足和欠缺经验的地方，非常感谢老师的耐心指导让我能如期完成这一份任务。跟随老师的指导进行开发让我学到了很多的东西，也让我对以后的项目开发有了一定的经验。

还要感谢辽宁省计量科学研究院给了我这次实践的机会，让我更早一步接触应用型研究开发，也非常感谢邢志红老师，凤伟老师在开发过程中的耐心解释与技术支持。

还有感谢开发小程序的兰淼同学的帮助，我在因私事落下的工作量，在她的帮助下短时间内补了回来。

最后，感谢全体答辩组的老师，感谢你们的付出。作为东北大学的学生，我感到无比的自豪。