

大连工业大学

# 实习报告

专    业：计算机科学与技术

实习类别：毕业实习

实习地点：北京泽乔文化传播公司

实习形式：分散实习

实习时间：2022/2/28—2022/3/11

外地 10 天、校内 0 天

学生姓名：梁宇龙

班级学号：1805010207

指导教师：唐晓君

## 目 录

第一章 实习相关说明 .....	- 1 -
1.1 实习目的 .....	- 1 -
1.2 实习要求 .....	- 1 -
1.3 实习日程 .....	- 1 -
第二章 实习内容 .....	- 2 -
2.1 实习单位与岗位简介 .....	- 2 -
2.2 实习内容与过程 .....	- 2 -
一、了解项目组开发目的及用户需求.....	- 2 -
二、学习 GIT 版本控制工具的使用 .....	- 2 -
三、学习 SOA 架构基本思想.....	- 3 -
四、学习 DUBBO 架构思想与服务注册中心 ZOOKEEPER 的使用 ....	- 3 -
五、模拟 DUBBO 在 SPRINGBOOT 项目中的使用和配置 .....	- 4 -
六、学习 DUBBO 管理控制台的使用 .....	- 4 -
七、学习 MAVEN 依赖传递和私服的使用 .....	- 4 -
八、了解研发团队组成和相关命名规范.....	- 5 -
九、“智慧食堂”项目的板块研发.....	- 5 -
十、模块交付说明会和开发文档的完善.....	- 6 -
2.3 实习总结与体会 .....	- 6 -

## 第一章 实习相关说明

### 1.1 实习目的

1. 具有计算机科学与技术领域的实习实践经历；
2. 熟悉计算机科学与技术领域相关的技术标准、知识产权、产业政策和法律法规；
3. 了解与本专业相关的重要法律、法规及方针与政策，并在实践中自觉遵守工程职业道德和规范，履行责任；
4. 能客观评价工程实践和复杂工程问题的解决方案对社会、安全、健康、法律以及文化的潜在影响，并理解应承担的责任；
5. 能够通过口头、书面、图表等形式与业界同行和人员进行有效沟通和交流；
6. 对本专业当前国际研究前沿与产业状况有基本了解。

### 1.2 实习要求

1. 实习期间必须服从指导教师的指导，学生按要求和规定严肃认真地完成实习任务。
2. 实习报告由大连工业大学提供标准格式，学生须按照指导教师要求认真书写。
3. 严格遵守作息时间，不迟到、不早退。
4. 自觉维护学校声誉，抵制不良倾向，团结友爱，爱护公共财物。
5. 严格遵守实习单位的各项规章制度，由于不遵守纪律而违反操作规程，造成事故或经济损失者，由当事人负责。

### 1.3 实习日程

表 1.1 实习日程

日期	实习内容
2 月 28 日	通过组会了解项目需求，学习 Git 版本控制工具的使用
3 月 1 日	了解 SOA 架构思想，学习 Apache Dubbo 工具使用
3 月 2 日	使用 Dubbo 工具整合 Maven 工程，学习服务注册中心使用和控制台监控
3 月 3 日	学习 Maven 的依赖传递解决方案，配置 Maven 私服
3 月 4 日	了解研发团队组成与命名规范，明确开发要求
3 月 7 日	组会布置本周工作任务，进行物联网端服务接口设计
3 月 8 日	解决接口设计中存在的权限控制问题，构建服务接口网关过滤器
3 月 9 日	解决接口设计中存在的跨域问题，重写@OriginCross 注解
3 月 10 日	开发读卡器数据提取服务，团队解决服务间的解耦问题
3 月 11 日	对本周个人开发内容进行文档编辑与修改，组会对开发部分进行说明

## 第二章 实习内容

### 2.1 实习单位与岗位简介

北京泽乔文化传播公司前身是泽乔新媒体工作室，工作室时期以凤凰网的运维和推广服务起家，2020 年成立公司。公司成立后开始拓宽新媒体推广业务和服务外包业务。研发团队共 73 人，以物联网业务为主要研发任务。实习期间所在研发团队为“某职业学校校园组网-智慧食堂”研发团队，本项目于 2021 年 4 月开始，预计于 2022 年 11 月完成验收，现进行功能模块开发阶段。实习期间所任岗位为 Java 开发工程师，负责在 SOA 架构下的的 Java 服务器端项目开发工作。

### 2.2 实习内容与过程

#### 一、了解项目组开发目的及用户需求

2 月 28 日，我进入项目组开始为期两周的实习任务。首先，项目经理带领我简单的介绍了项目背景。本次项目是公司与北京市某职业学校合作将该校新建的食堂窗口支付功能整合到该校学生统一支付平台（一卡通）中，项目总体需求为实现学生使用一卡通在学校食堂进行支付和扣费，校方通过一卡通消费记录对食堂承包商进行结算，食堂承包商可掌握各窗口刷卡机流量数据与校方核实结算金额。根据项目经理安排，首先会根据当前项目整体设计中需要的技术知识对我进行培训，并学习版本控制工具的使用。在培训结束后，通过组会中的相关安排，分配相应开发工作。

#### 二、学习 Git 版本控制工具的使用

2 月 28 日组会后，项目经理向我介绍了在小组中解决团队中代码冲突的解决方案——Git 版本控制工具。Git 是分布式版本控制系统，分为本地仓库和远程仓库，本地仓库是开发人员在自己电脑上的 Git 仓库，远程仓库是在远程服务器上的 Git 仓库。

作为版本控制工具，Git 有两个非常重要的概念，即分支（Branch）和标签（Tag）。根据项目经理介绍，当进入小组工作时，首先将已经通过测试的原有工程文件克隆到本机，随后创建一个分支，确保在自己开发过程中，原有工程文件的迭代不会受到影响。当开发工作进行到一个阶段性成果（如某方法通过测试，某服务已经完成工程整合）的情况下，在自己工程的分支加入标签，并记录标签完成的阶段性功能。通过使用标签，可以在某功能开发出现问题的情况下，退回到标签记录的状态，不会因相应配置或依赖影响程序运行。当开发任务完成并交付测试组完成测试后，经项目经理批准，将分支推送到远程仓库，并由项目经理

完成分支的合并，由于在创建分支后，可能主分支会继续添加新功能模块，导致出现代码冲突，此时，项目经理会召开组会讨论，并提供冲突代码的解决方案。

在项目经理介绍完成后，于 2 月 28 日下午，我开始进行自主学习 Git 相应命令和使用方法，在小组长的帮助下，在码云（Gitee）平台将我加入团队，并将 Git 测试项目和开发文档交由我进行实验。实验中主要体验了在本地主机中文件的三种状态之间的转换，并根据文档验证在两个不同分支下对工程进行不同操作后，验证合并分支过程中的代码冲突的表现形式。

3 月 1 日组会中，检查上周完成测试代码分支合并过程中的冲突问题，根据冲突显示，在大多数情况下代码冲突为配置项的冲突，即响应配置等方面的配置项发生冲突，各小组在配置项对小组程序影响说明后，项目经理与各小组协商归并冲突，并将冲突修改后的工程打标签交由集成测试。

### 三、学习 SOA 架构基本思想

3 月 1 日组会后，项目经理向我安排今日学习任务，即 SOA 架构思想与 Apache Dubbo 工具的使用练习。首先项目经理向我讲述了项目工程架构的发展历程，讲述了单体架构与垂直架构的局限性是功能集中性强，不利于开发维护，进而引出 SOA 架构的解决方案。

SOA 架构全称为 Service-Oriented Architecture，即面向服务的架构。在该架构中，可以通过网络（ESB 服务总线）对低耦合的服务分布式部署和组合，每个服务以独立的形式存在于操作系统进程中。通过这种架构可以将重复模块抽取为服务，提高开发效率，有较高重用性和维护性。

### 四、学习 Dubbo 架构思想与服务注册中心 Zookeeper 的使用

介绍 SOA 架构后，项目经理向我简单介绍了 Apache Dubbo 工具，Dubbo 是一款高性能的 Java RPC 框架，可以与 Spring 框架无缝集成。RPC 全称为 remote procedure call，即远程过程调用，可以实现两个应用部署在两个服务器上，可以通过 RPC 可以实现调用。

解释完 RPC 后，开始进入 Dubbo 架构的学习。Dubbo 架构实现调用的核心组件是 Registry，即服务注册中心（服务容器）。在架构中，服务注册中心负责启动、加载、运行服务提供者。服务提供者在启动时，向注册中心注册自己提供的服务；服务消费者在启动时，向注册中心订阅自己的服务。注册中心返回服务提供方的地址列表给消费者，在消费者方会基于负载均衡算法选择提供者的一个地址进行调用，同时注册中心会监控调用次数与时间。

在项目经理讲解结束后，通过下发的文档自学服务注册中心解决方案 Zookeeper 的接入方式并模拟使用 dubbo 作为服务提供者接入 Zookeeper 获取服

务。

## 五、模拟 Dubbo 在 SpringBoot 项目中的使用和配置

根据文档学习和自主实验，对于 Dubbo 接入服务提供方工程的相关总结如下：首先作为 maven 工程，需要在 pom.xml 文件中引入 dubbo、zookeeper、zkclient 和 fastjson 坐标依赖，由父工程（项目团队）管理版本。在团队开发中，对暴露服务的方法，需要以实现接口的形式进行开发，以便于团队中其他使用者直接使用接口完成相应功能。完成服务接口和实现类开发后，需要在 application.yml 文件中配置 dubbo，主要参数为：

application: 配置属性 name 为当前应用的名称，用于注册中心计算应用依赖关系。相关命名需要在组会中由项目经理进行分配，便于使用服务的人正确找到相应服务。

registry: 配置属性 address 为 zookeeper 所在服务器的 IP 地址和端口号。

protocol: 注册协议和端口，配置 name 表明使用 dubbo 协议，配置 port 为当前服务所在端口。

annotation: 配置 package 属性为扫描包，在包内加入@Service 注解的类会被发布为服务。

当运行 SpringBoot 项目后，编写好的服务类就正式被发布。通过 Dubbo 管理控制台即可查看。

## 六、学习 Dubbo 管理控制台的使用

在上述实验运行结束后，项目经理安排我去到需要用到服务订阅的团队学习调用方法。负责人首先和我介绍服务订阅方在 maven 工程中的使用，主要通过 @Reference 注解在服务注册中心获取发布的服务类并依赖注入，随后直接使用该对象方法即可完成调用。当然前提是连接了服务注册中心。

由于订阅方需要根据提供方对具体功能实现进行接口连接，因此该团队中部署了管理中心。管理中心实际上是一个 web 应用，将 war 文件部署到 tomcat 上便可使用，但需要在启动 tomcat 后在 WEB-INF 的 properties 文件中修改 zookeeper 的 IP 地址、端口号、用户名、密码。通过访问路径，输入用户名密码后便可观察到已发布的服务和订阅的用户。

## 七、学习 Maven 依赖传递和私服的使用

在学习完成 Dubbo 的相应使用后，为了让团队更好地实现对相应依赖的版本控制，项目小组组长向我介绍了 Maven 的依赖传递和私服使用。首先，组长假设存在三个项目，项目 C 依赖项目 B，B 依赖 A，则 C 也要依赖 A，就构成了依赖传递。如 Spring-webmvc 依赖 Spring-beans-4.2.4，Spring-aop 依赖

Spring-beans-5.0.2, 假设 4.2.4 已经加入工程中, 而希望 5.0.2 加入工程, 就造成了依赖冲突。

解决依赖冲突的方法分别是: 依赖调节原则、排除依赖、锁定版本。

依赖调节原则的两种方式时第一声明者优先原则和路径近者优先原则。第一声明者即 Maven 中以先声明的依赖为准, 根据坐标导入的顺序决定使用哪个依赖; 排除依赖可以使用<exclusions>标签排除传递过来的依赖; 版本锁定可以在<dependencyManagement>标签中锁定依赖的版本, 锁定后不考虑依赖调节原则, 已锁定的版本为准。

在本项目组内的父工程中以锁定依赖版本, 从 Gitee 仓库拉取后在子工程中则不需要填写版本。若有版本冲突问题, 需要及时与团队沟通。

之后, 组长向我介绍在 Maven 工程中, 公司具有自己的私服仓库。当完成某个服务后, 可以将服务打成 jar 包发到私服中, 其他成员如果需要用到, 就可以以依赖形式引入你写好的服务。同时私服也充当一个代理服务器的角色, 当私服没有 jar 包是, 还是要从 Alibaba 中央仓库镜像中下载。随后通过文档练习了项目发布到私服和下载至本地仓库。

## 八、了解研发团队组成和相关命名规范

本项目在公司内共分为八个小组, 除去物联网设计小组、管理系统前端小组和测试组之外, 共有五个小组为后端服务开发。我所在的小组需完成数据库接口设计以及根据物联网接口提供请求对数据库进行增、改、查操作。在第二周实习中我需要完成物联网接口设计和服务调用的连接。

在开发之前, 项目经理将本项目设计文档交付于我, 上面严格说明了开发过程中的所有命名规范。如包命名规则为: (公司名. 团队名. 小组名. 开发者); 若需要编写配置文件时需要在自编写部分前后加入注释并表明添加者姓名和所在团队; 所有依赖修改必须在组会中提出, 由项目经理统一调度; 完成服务设计后需创建分支后再上传至码云交由测试; 接口设计中 RequestMapping 的一级路径写明开发者小组, 接口需加上中英文注释, 类内变量必须设置静态等一系列标准。

最后, 项目经理强调了代码保护和开发环境授权问题。主要强调代码不可带出公司、使用开源开发环境 (建议使用 VS Code)、不能个人设备连接私服和 zookeeper 等一系列安全说明。

## 九、“智慧食堂”项目的板块研发

第二周实习中, 我进行了为期四天的模块开发, 根据上周的工作安排, 本周需要根据开发文档设计并实现物联网接口, 帮助物联网小组调用相关服务, 并完成一个自己负责的服务。周一组会结束后便开始工作, 相比于上周的培训, 本周

开发时间相对松弛。但相比于在学校进行项目研发，不同的是，在开发服务结束后，需要严格的书写开发文档，只有写明开发文档后，才能将服务模块交由测试组测试。

首先是设计物联网端接口，根据开发文档提供的接口设计说明将接口设计并传入服务中心。在接口设计过程中遇到的最主要问题就是服务器的跨域请求。SpringBoot 的依赖中内置了 tomcat 服务器，并可在 yml 中配置端口号。为了避免分布式部署，端口号在设计文档中都有了明确说明。但在第二天测试组发来的测试报告中显示，当使用本地客户端访问服务器中服务时，会报出 500 的错误，并说明存在跨域的问题。根据这个问题，组长向我讲述了解决方案。在管理系统中如果面向前端工程化项目，如 Angular/Vue 项目时，需要由前端设置代理服务器解决跨域问题；而面向物联网等配置较难的接口时，更多的是在 Controller 中配置 @OriginCross 注解，标注允许跨域请求该服务。

接口设计完成后，我便开始了服务的开发工作。本次我的开发工作是当物联网端读取一卡通信息后，会在接口中传递卡片 ID、读卡器 ID 参数，可能传递消费金额参数。当未传递消费金额参数时，需要回传给物联网端一卡通储值金额；当传递消费参数时，需要回传给读卡器剩余金额。在开发过程中遇到的问题主要是解耦，在最开始的开发过程中，我需要调用其他已经写好的服务，而我要写的服务也是需要其他服务调用使用的，就导致我的服务中重现了多个服务调用过程，除了配置项和服务名过于复杂外，维护也非常麻烦。通过组长和团队讨论，做出了解耦的设计，通过将一个服务中多重条件判断放置在接口的路由重定向实现，将不同的情况拆分成多个不同的服务，将多个服务中都需要的代码抽出为公共服务调用，比较好的完成了该模块的设计。但在 SOA 架构下，只通过一个总线的开发方式局限性仍然很大，但对该项目的并发，足以应对。

## 十、模块交付说明会和开发文档的完善

实习最后一天时，首先根据测试组发来的测试报告修改了相应的服务内容后，在离职前向团队中开展模块交付说明会。说明会中将服务完成了什么功能，需要什么参数，调用方式和跨域解决办法的接口进行了相应说明，并回答了关于代码复用问题的相关解决办法。随后，团队审阅了我的开发文档，根据组长和项目经理提出的问题，对相应文档说明进行调整后正式办理离职，并感谢项目组和其他同学的照顾。

## 2.3 实习总结与体会

本次实习是我从学校跨入实际生产环境的第一次实验。通过本次实习了解了真实的生产环境和校园实验实训开发环境的不同，技术的使用方式以及生产环境



下代码开发的注意点，为接下来的学习和工作在生产环境下的了解打下了一定基础。接下来，我将从生产环境、技术特点和工程伦理三个方向总结本次实习。

在生产环境中，由于在校期间大多数时候都是自己完成的微型项目，在完成相应逻辑设计和模块开发后使程序能够正确运行即是完成任务。而在正式的生产环境中，除了保证代码的正确性以外，权限鉴别、反爬虫、安全保护、代码注入等许多方面都是需要注意的，保证代码安全和保证代码正确性同等重要。此外就是生产环境中由于是团队工作机制，因此在代码暴露的接口、服务名等方面，不仅仅是要让自己看懂，让别人看懂也非常重要。通过本次实习中的项目设计，无论是在接口代码上的中英文说明，还是开发文档中的模块服务解释，都不是在学校中简述即可完成。

技术特点上，单体的 Spring Boot 项目在如今的项目并发体系下已经无法完成高并发的要求。取而代之的 SOA 架构和微服务架构，如何完成高并发配置成为了非常重要的思考问题。我曾经以为服务器端开发调取数据库数据是再简单不过的，而在分布式架构中，数据库的权限判定和服务调用的接口配置都是需要注意且仔细设计的。

工程伦理上，例如团队开发包名和接口名设计已经是很正常的事情了。但除此之外，服务复用性、服务使用的权限判定等问题，代码冲突后如何解决各部门之间的协调配置问题，分布式负载均衡问题下的算法调配问题仍是需要仔细钻研设计的。此外，无论是作为学生还是工作者，保护自己和公司的代码产权利益还是要摆在首位的，不把代码带出公司，不私自连接公司私服等问题，需要每一个行业工作者时刻牢记。

通过本次实习，与其说检验学习成果，不如说是去外面见见世面。即使大学四年很系统的学过理论知识，即使觉得自己的逻辑无懈可击，但在一个大型项目摆在面前，我们作为学生考虑问题的方式仍如井底之蛙一样狭隘。也正是通过本次实习中学习到的相关知识，在接下来的毕业设计中，思考问题的方式会变的更加宽泛，为毕业设计更接近生产环境提供了理论基础。