

生产实习报告

**专业班级：**计算机科学与技术1605班

**学 号：** U201614633

**姓 名：** 李超峰

**指导教师：** 谢夏

**实习单位：** 华中科技大学

**实习时间：** 2019.07.01-2019.09.1

**计算机科学与技术学院**

目录

[1 实习概要 1](#_Toc21368559)

[1.1 实习目的 1](#_Toc21368560)

[1.2 实习内容 1](#_Toc21368561)

[1.2.1 调查和记录 1](#_Toc21368562)

[1.2.2 生产实践 2](#_Toc21368563)

[1.3 实习单位及实习岗位简介 2](#_Toc21368564)

[1.3.1 实习单位 2](#_Toc21368565)

[1.3.2 实习指导老师简介 4](#_Toc21368566)

[1.3.2 实习岗位 5](#_Toc21368567)

[1.3.3 实习计划 5](#_Toc21368568)

[2 实习报告 6](#_Toc21368569)

[2.1 发展概述 6](#_Toc21368570)

[2.2 基本原理分析 8](#_Toc21368571)

[2.2.1 MVC原理 8](#_Toc21368572)

[2.2.2 SpringMVC框架 9](#_Toc21368573)

[2.2.3 Spring IOC和AOP 11](#_Toc21368574)

[2.2.3 Spring boot整合mybatis框架 13](#_Toc21368575)

[2.2.4 Spring boot整合Redis 15](#_Toc21368576)

[2.3 实习内容 16](#_Toc21368577)

[2.3.1 阅读相关文献 16](#_Toc21368578)

[2.3.2 JAVA环境搭建以及IDEA下载 17](#_Toc21368579)

[2.3.3 Springboot web代码学习 20](#_Toc21368580)

[2.4 实习工作总结 33](#_Toc21368581)

[2.5 实习建议 33](#_Toc21368582)

[3 实习问题解答 34](#_Toc21368583)

[4 生产实习指标点以及问题回答 36](#_Toc21368584)

[参考文献 38](#_Toc21368585)

# 1 实习概要

## 1.1 实习目的

生产实习是高等工科院校教学计划中实践教学的重要环节之一，是理论与实际相结合的有效方式。

通过生产实习环节，学生可以利用所学的理论知识去分析实习中所看到的实际生产技术，从而将科学的理论知识加以验证、深化、巩固和充实。生产实习还能培养学生进行调查、研究、分析和解决实际问题的能力，扩展学生的知识面，既体会到学习书本知识的必要性，又提高解决实际工程问题的能力。此外，生产实习也是学生接触社会、了解社会的重要途径。

为后续专业课程的学习、课程设计和毕业设计打下坚实的基础。通过生产实习，还可以拓宽学生的知识面，增加感性认识，把所学知识条理化、系统化，学习一些从书本上学不到的专业知识，并获得本专业国内、国外科技发展现状的最新信息，激发学生向实践学习和探索的积极性，为今后的学习和将从事的技术工作打下坚实的基础。

培养学生工程技术素养、实现理论联系实际的重要措施。通过生产实习，是学生把所学的理论知识与生产实际相结合，重点培养学生分析问题和解决实际问题的能力。

## 1.2 实习内容

生产实习内容包括两个方面：

1. 调查和记录
2. 生产实践

### 1.2.1 调查和记录

1. 调查和记载实习单位的管理方式和经营情况，包括：

· 单位的组织结构

· 技术部门结构及人员配备情况

· 生产的产品或销售规模、产值、产量及新产品的研制情况

· 实习单位的各种规章制度，包括规章制度名称和重要规定

2. 记录产品研制、开发的方式及过程，包括：

· 计算机硬件、软件的组成、结构和性能、测试机维修等情况，用学过的表示方法加以描述、绘制和记载

· 记载产品生产和装配的工序、工艺及各个环节上的重要处理技术

· 产品开发或研制的技术特点及技术性能的先进性

3. 收集整理产品开发的技术资料，包括：

· 硬部件的线路图及技术参数

· 各种软件设计文档

· 产品性能指标及参数

· 产品使用说明书

以上内容应严格按照生产实习单位的工作要求有选择地记载，具体内容可包括各种资料的类型、格式、图样及全部或部分内容，主义生产实习单位保密工作要求，所记载内容均写在“生产实习日记”中。

### 1.2.2 生产实践

根据所在实习单位的具体条件，完成下列一至多项工作并作详细记载：

1. 了解计算机、计算机网络和计算机新技术的实际应用情况

2. 熟悉或参加某个软件项目的开发工作，包括：软件开发方法、软件开发工具、软件项目管理及文档资料整理工作。

3. 参与系统总体调试工作

4. 掌握软件开发工具及多媒体软件的应用

5. 了解计算机网络的体系结构，高层功能分化以及新技术的应用。

## 1.3 实习单位及实习岗位简介

### 1.3.1 实习单位

华中科技大学大数据课题组，隶属于华中科技大学服务计算技术与系统实验室与集群与网络计算实验室

华中科技大学“服务计算技术与系统教育部重点实验室”暨“集群与网格计算湖北省重点实验室”依托于计算机系统结构国家重点学科和计算机软件与理论湖北省重点学科，建有湖北省大数据技术与系统工程实验室，拥有自由开放的学术氛围和国际前沿的研究方向。目前主要的研究领域包括：系统软件与体系结构、分布式系统、网络空间安全、大数据等。

实验室具备实力雄厚的师资力量、充满活力的科研梯队以及良好的硬件设施环境。现有教授11人，副教授17人，具有博士学位者29人，其中长江学者特聘教授1人，973计划项目首席科学家2人次、国家杰出青年基金获得者1人、“新世纪百千万人才工程”国家级入选1人、湖北省高端人才引领培养计划入选1人、全国百优1人、教育部“长江学者”青年项目获得者1人、中组部“青年拔尖人才支持计划” 入选2人、国家自然科学基金优秀青年基金获得者3人、“新世纪优秀人才支持计划”入选3人、“楚天学子”入选2人、湖北省杰出青年基金获得者3人、湖北省十佳师德标兵1人、武汉市“青年科技晨光计划”入选4人、微软亚洲研究院“铸星计划”3人。目前在读全日制博士、硕士研究生200余人。实验室承建了中国教育科研网格ChinaGrid主结点、中国国家网格CNGrid（武汉）结点、985科技创新平台；拥有3000平方米实验基地，主要实验设备资产总值达9千多万元。实验室发展与建设将直接为高层次人才的培养提供良好的基础设施与外部条件。

实验室的主要学术方向及研究内容有：

（1）系统软件与体系架构：利用新型处理器和新型存储器件为上层新型应用提供有效的支撑平台，包括：1）面向大数据处理的体系结构，主要研究内存计算和图计算机体系结构相关的问题；2）编程模型和运行时优化技术，力求为海量数据环境下的复杂计算模型建立简单、高效的并行编程模型和运行时系统，并探索流处理、迭代处理等模型的优化技术；3）系统虚拟化技术，研究具体物理部件的虚拟化技术、嵌入式系统的虚拟化技术和桌面虚拟化技术等。

（2）分布式系统：本研究方向定位于云计算、数据中心、云端融合、区块链等分布式处理领域，研究内容包括云操作系统与容器虚拟化、数据中心资源管理与绿色计算、软件定义网络与网络虚拟化、边缘计算与新型云端融合架构等，同时利用云计算技术在工程计算、视频处理、制造服务、智慧交通与智能家电等应用领域开展了一系列云应用。

（3）大数据：研究大数据基础理论、大数据处理、大数据管理与大数据分析等问题。主要包括：1）大数据的新型算法理论，主要研究设计严格亚线性复杂度分布式算法以及在并行分布式环境下对大数据处理问题难度的精细划分；2）大数据处理系统，侧重于新型软硬件架构上的系统优化，如内存计算、异构计算、流计算等；3）大数据管理系统，重点研究图计算与图数据库、知识管理、大数据管理查询等；4）大数据分析，主要针对大规模数据的有效机器学习算法的设计与实现，以及对海量数据的有效信息进行挖掘，并构建典型系统等。

（4）网络空间安全：主要研究1）云计算安全，针对动态复杂的云计算环境，从云系统、云服务和云数据三个层面开展关键安全理论和技术的研究；2）软件安全，开展软件的错误检测和漏洞挖掘研究；3）区块链技术，研究共识机制安全、智能合约安全、数字资产安全以及区块链应用。

实验室承担了70余项重大科研项目，包括国家973项目、教育部重大专项、国家科技支撑计划项目、国家科技重大专项、国家杰出青年基金项目、国家自然科学基金重大/重点项目、国家863重大/重点项目、国家发改委CNGI项目、科技部国际合作项目等。实验室现为科技部重点领域创新团队、教育部“长江学者和创新团队发展计划”创新团队牵头单位、湖北省自然科学基金创新团队。

实验室出版专著和教材18余本，发表国内外学术期刊论文及国际学术会议论文近800篇，获得国际发明专利6项、获得国家发明专利201项，在审国家发明专利127项，获得国家软件著作版权162项；获得国家科技进步二等奖2项、国家技术发明二等奖1项、国家自然科学四等奖1项、省部级科技进步/技术发明一等奖4项、二等奖1项、湖北省教学成果一等奖1项。

实验室坚持开放与联合，与美国、德国、澳大利亚、日本、英国、法国、加拿大、匈牙利、挪威等国家和香港、台湾地区的大学，以及Intel、惠普、微软、IBM、AMD、法国电信、曙光、UT斯达康、华为、中国移动、腾讯、天涯在线、阿里巴巴、海尔、西门子等国内外知名IT企业保持着密切合作，先后建立了华中科技大学-惠普高性能计算联合实验室、Intel“安腾”应用研发中心、UT斯达康IPTV联合实验室、Intel“多核技术”实验室、华中科技大学-天涯在线联合实验室、阿里巴巴联合实验室、华中科技大学—腾讯联合实验室。此外，实验室成功举办了近30余次具有重要影响的国际/国内学术会议。

实验室本着“明德厚学、求是创新”的华中大精神，坚持“育才明德、开放竞争”的原则，不断开拓进取，致力成为国内一流、国际知名的研究开发基地和人才培养基地。

### 1.3.2 实习指导老师简介

谢夏，博士，华中科技大学计算机学院副教授。主要学术方向为数据挖掘、性能分析等。先后主持或参与国家自然科学基金、国家重点研发专项、支撑计划等多个项目。在System Journal、Information Fusion等国际会议与期刊发表论文多篇，是IEEE/CCF 会员。

主要的研究方向：并行计算、性能评估、数据挖掘等

### 1.3.2 实习岗位

大数据项目组之web前后端的开发，该组由一名硕士学长李璇带领，4名实习生共同参与。主要工作是学习了解Springboot框架与mybatis , redis,seesion, security等相关技术的整合的以及支付模块的书写

### 1.3.3 实习计划

（1）第一阶段：基础知识学习

阅读、学习关于java语言程序设计的基础知识，springboot的相关知识，web先后端设计以及前后端交互的基础知识，以及微信支付相关文档。

（2）第二阶段：安装配置IDEA

从官网上面下载IDEA

（3）第三阶段：阅读理解和实践代码

下载相关的代码，阅读代码，然后实战，自己动手做一些简单的web开发，了解web开发的大体流程。

（4）第四阶段：探讨交流

实习生交流web开发的收获与心得，探讨Java web的实现原理。

# 2 实习报告

## 2.1 发展概述

**1. Spring boot**

Spring框架是Java平台上的一种开源应用框架，提供具有控制反转特性的容器。尽管Spring框架自身对编程模型没有限制，但其在Java应用中的频繁使用让它备受青睐，以至于后来让它作为[EJB](https://baike.baidu.com/item/EJB/144195" \t "_blank)（EnterpriseJavaBeans）模型的补充，甚至是替补。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。Spring框架具有控制反转（IOC）特性，IOC旨在方便项目维护和测试，它提供了一种通过Java的反射机制对Java对象进行统一的配置和管理的方法。Spring框架利用容器管理对象的生命周期，容器可以通过扫描XML文件或类上特定Java注解来配置对象，开发者可以通过依赖查找或依赖注入来获得对象。Spring框架具有[面向切面编程](https://baike.baidu.com/item/%E9%9D%A2%E5%90%91%E5%88%87%E9%9D%A2%E7%BC%96%E7%A8%8B/6016335" \t "_blank)（AOP）框架，SpringAOP框架基于代理模式，同时运行时可配置；AOP框架主要针对模块之间的交叉关注点进行模块化。Spring框架的AOP框架仅提供基本的AOP特性，虽无法与AspectJ框架相比，但通过与AspectJ的集成，也可以满足基本需求。Spring框架下的事务管理、远程访问等功能均可以通过使用SpringAOP技术实现。Spring的事务管理框架为Java平台带来了一种抽象机制，使本地和全局事务以及嵌套事务能够与保存点一起工作，并且几乎可以在Java平台的任何环境中工作。Spring集成多种事务模板，系统可以通过事务模板、XML或Java注解进行事务配置，并且事务框架集成了消息传递和缓存等功能。Spring的数据访问框架解决了开发人员在应用程序中使用数据库时遇到的常见困难。它不仅对Java:JDBC、iBATS/MyBATIs、Hibernate、Java数据对象（JDO）、ApacheOJB和ApacheCayne等所有流行的数据访问框架中提供支持，同时还可以与Spring的事务管理一起使用，为数据访问提供了灵活的抽象。Spring框架最初是没有打算构建一个自己的WebMVC框架，其开发人员在开发过程中认为现有的StrutsWeb框架的呈现层和请求处理层之间以及请求处理层和模型之间的分离不够，于是创建了SpringMVC

Spring Boot是一个基于Java的开源框架，用于创建微服务。它由Pivotal Team开发，用于构建独立的生产就绪Spring应用。

微服务(Micro Service)是一种允许开发人员独立开发和部署服务的体系结构。每个运行的服务都有自己的流程，这实现了轻量级模型以支持业务应用程序。

SpringBoot所具备的特征有：

（1）可以创建独立的Spring应用程序，并且基于其Maven或Gradle插件，可以创建可执行的JARs和WARs；

（2）内嵌Tomcat或Jetty等Servlet容器；

（3）提供自动配置的“starter”项目对象模型（POMS）以简化Maven配置；

（4）尽可能自动配置Spring容器；

（5）提供准备好的特性，如指标、健康检查和外部化配置；

（6）绝对没有代码生成，不需要XML配置

SpringBoot框架中还有两个非常重要的策略：开箱即用和约定优于配置。开箱即用，Outofbox，是指在开发过程中，通过在MAVEN项目的pom文件中添加相关依赖包，然后使用对应注解来代替繁琐的XML配置文件以管理对象的生命周期。这个特点使得开发人员摆脱了复杂的配置工作以及依赖的管理工作，更加专注于业务逻辑。约定优于配置，Convention over configuration，是一种由SpringBoot本身来配置目标结构，由开发者在结构中添加信息的软件设计范式。这一特点虽降低了部分灵活性，增加了BUG定位的复杂性，但减少了开发人员需要做出决定的数量，同时减少了大量的XML配置，并且可以将代码编译、测试和打包等工作自动化。

SpringBoot应用系统开发模板的基本架构设计从前端到后台进行说明：前端常使用模板引擎，主要有FreeMarker和Thymeleaf，它们都是用Java语言编写的，渲染模板并输出相应文本，使得界面的设计与应用的逻辑分离，同时前端开发还会使用到Bootstrap、AngularJS、JQuery等；在浏览器的数据传输格式上采用Json，非xml，同时提供RESTfulAPI；SpringMVC框架用于数据到达服务器后处理请求；到数据访问层主要有Hibernate、MyBatis、JPA等持久层框架；数据库常用MySQL；开发工具推荐IntelliJIDEA

## 2.2 基本原理分析

### 2.2.1 MVC原理

MVC是一种常见的设计模式，很多框架都支持MVC这种模式，springboot框架也同样支持这种模式。

MVC原理图如下所示：

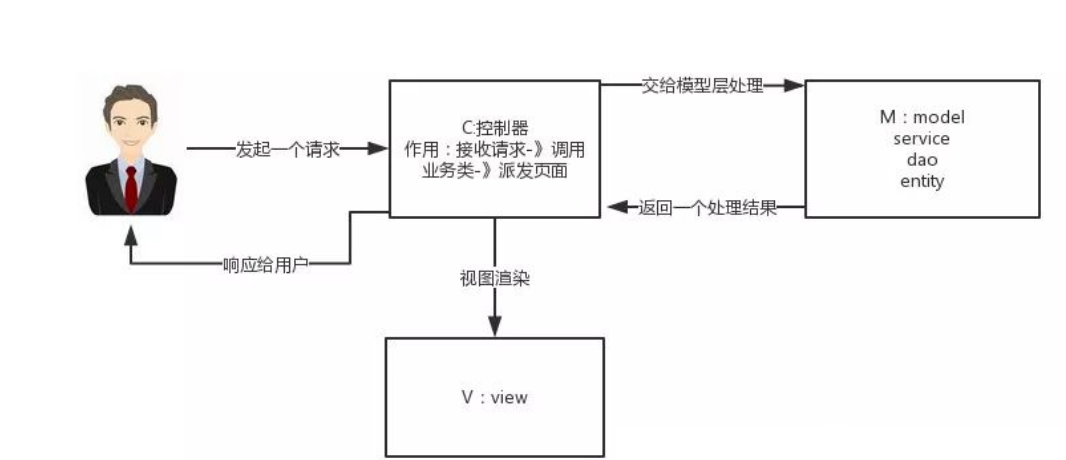


图2.1

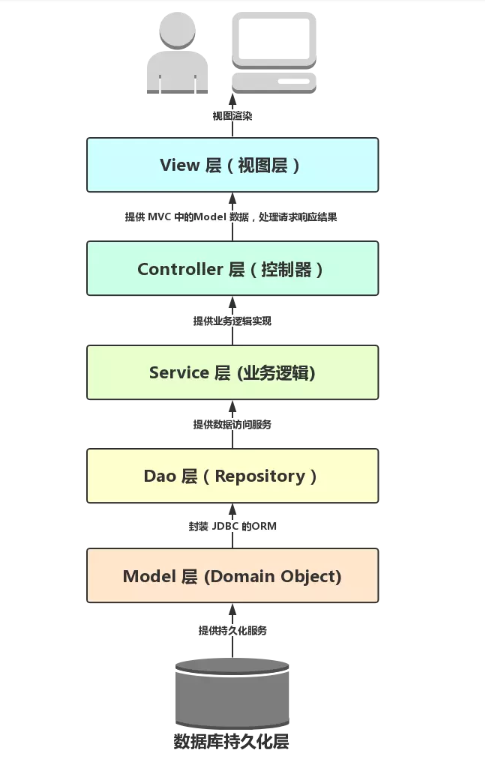
MVC的具体指向：

M-Model 模型（完成业务逻辑：service+dao+entity）

V-View 视图（做界面的展示 jsp，html……）

C-Controller 控制器（接收请求—>调用模型—>根据结果派发页面）

大多数的企业级开发都会用到MVC模式，J2EE常见的架构图如下所示：



### 2.2.2 SpringMVC框架

SpringMVC是一个MVC的开源框架，springMVC就相当于是Struts2与Sring框架的整合，SpringMVC是Spring框架的一个后续产品（一个模块），其实就是Spring在原有基础上，又提供了web应用的MVC模块，可以简单的把SpringMVC理解为是Spring的一个模块。

**SpringMVC的系统框架如图所示:**

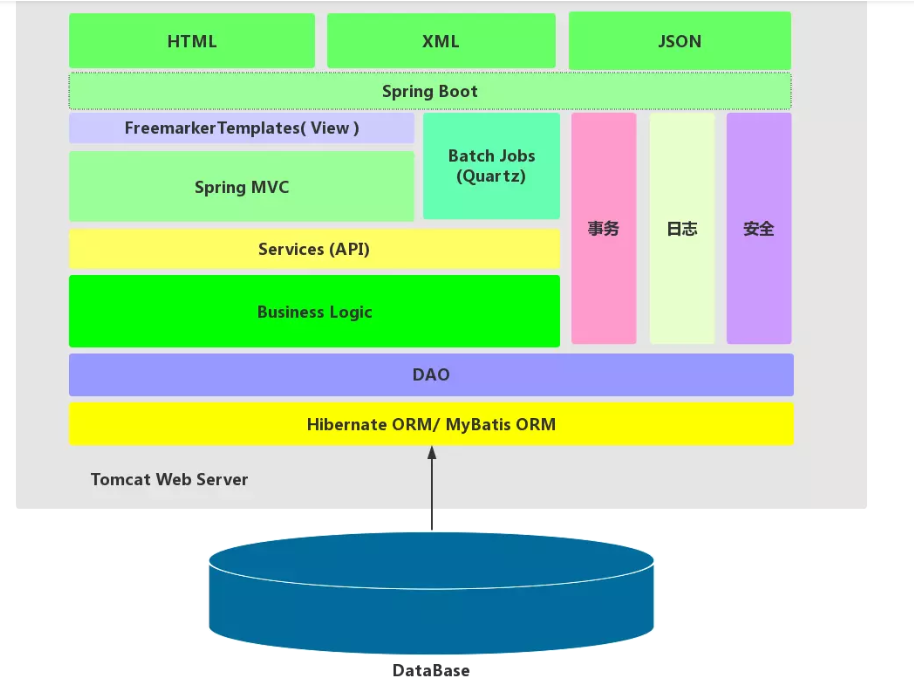


图2.3

**SpringMVC的具体工作原理如下：**

1、 客户端用户发送请求至前端控制器DispatcherServlet。

2、 DispatcherServlet收到请求调用HandlerMapping处理器映射器。

3、 HandlerMapping处理器映射器找到具体的处理器(可以根据xml配置、注解进行查找)，生成处理器对象及处理器拦截器(如果有则生成)一并返回给DispatcherServlet。

4、 DispatcherServlet调用HandlerAdapter处理器适配器，HandlerAdapter经过适配调用具体的处理器(Controller，也叫后端控制器)。

5、 Controller控制器执行完成返回ModelAndView，HandlerAdapter将controller执行结果ModelAndView返回给DispatcherServlet。

6、 DispatcherServlet前端控制器将ModelAndView传给ViewReslover视图解析器。

7、 ViewReslover解析后返回具体View。

8、 DispatcherServlet根据View进行渲染视图（即将模型数据填充至视图中），并返回给客户端用户。

**SpringMVC框架组件说明**：

以下组件通常使用框架提供实现：

DispatcherServlet：作为前端控制器，整个流程控制的中心，控制其它组件执行，统一调度，降低组件之间的耦合性，提高每个组件的扩展性。

HandlerMapping：通过扩展处理器映射器实现不同的映射方式，例如：配置文件方式，实现接口方式，注解方式等。

HandlAdapter：通过扩展处理器适配器，支持更多类型的处理器。

ViewResolver：通过扩展视图解析器，支持更多类型的视图解析，例如：jsp、freemarker、pdf、excel等。

**SpringMVC框架的工作原理图：**

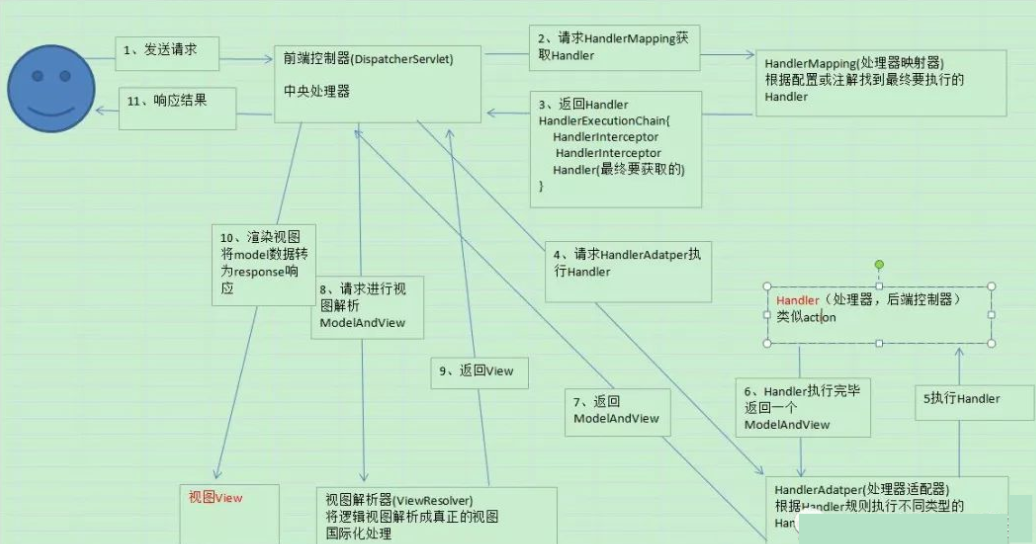
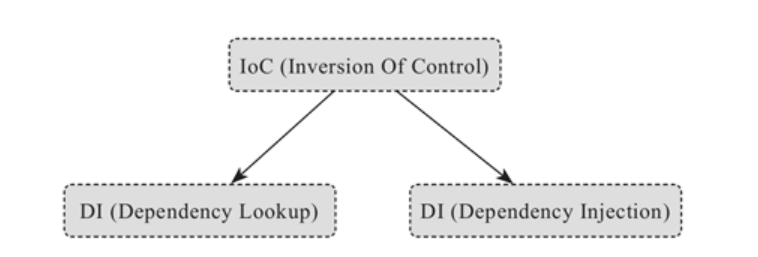


图2.4

### 2.2.3 Spring IOC和AOP

Springboot框架内部最核心的就是IOC了，动态注入，让一个对象的创建不用new了，可以自动的生产，这其实就是利用java里的反射，反射其实就是在运行时动态的去创建、调用对象，Spring就是在运行时，跟xml Spring的配置文件来动态的创建对象，和调用对象里的方法的。概念之间的关系如图 1 所示



Spring还有一个核心就是AOP这个就是面向切面编程，可以为某一类对象 进行监督和控制（也就是 在调用这类对象的具体方法的前后去调用你指定的 模块）从而达到对一个模块扩充的功能。这些都是通过 配置类达到的。

Spring目的：就是让对象与对象（模块与模块）之间的关系没有通过代码来关联，都是通过配置类说明管理的（Spring根据这些配置 内部通过反射去动态的组装对象） 要记住：Spring是一个容器，凡是在容器里的对象才会有Spring所提供的这些服务和功能。

**Spring AOP与IOC**

一、 IoC(Inversion of control): 控制反转

1、IoC：

概念：控制权由对象本身转向容器；由容器根据配置文件去创建实例并创建各个实例之间的依赖关系

核心：bean工厂；在Spring中，bean工厂创建的各个实例称作bean

二、AOP(Aspect-Oriented Programming): 面向方面编程

1、 代理的两种方式：

静态代理：

针对每个具体类分别编写代理类；

针对一个接口编写一个代理类；

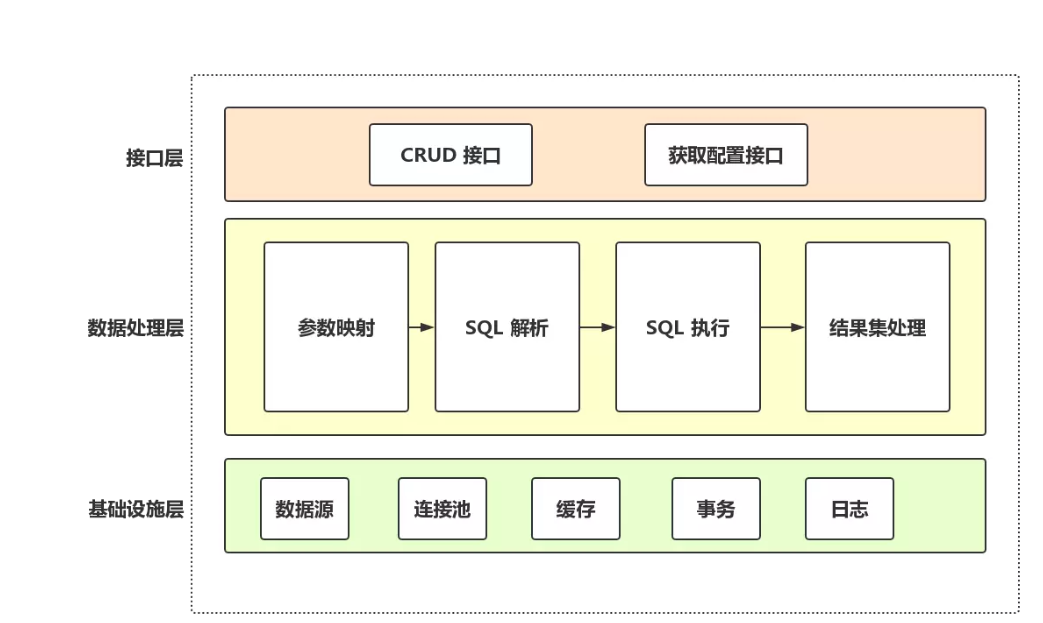
动态代理：

针对一个方面编写一个InvocationHandler，然后借用JDK反射包中的Proxy类为各种接口动态生成相应的代理类

### 2.2.3 Spring boot整合mybatis框架

MyBatis 是 Apache 的一个开源项目——iBatis，2010 年这个项目由 Apache Software Foundation 迁移到了 Google Code，并且改名为 MyBatis。MyBatis 是一个基于 Java 的持久层框架。MyBatis 提供的持久层框架包括 SQL Maps 和 Data Access Objects（DAO），它消除了几乎所有的 JDBC 代码和参数的手工设置以及结果集的检索。MyBatis 使用简单的 XML 或注解用于配置和原始映射，将接口和 Java 的 POJOs（Plain Old Java Objects，普通的 Java 对象）映射成数据库中的记录。

MyBatis的架构图如下所示：



**MyBatis 的工作原理**

1）读取 MyBatis 配置文件：mybatis-config.xml 为 MyBatis 的全局配置文件，配置了 MyBatis 的运行环境等信息，例如数据库连接信息。

2）加载映射文件。映射文件即 SQL 映射文件，该文件中配置了操作数据库的 SQL 语句，需要在 MyBatis 配置文件 mybatis-config.xml 中加载。mybatis-config.xml 文件可以加载多个映射文件，每个文件对应数据库中的一张表。

3）构造会话工厂：通过 MyBatis 的环境等配置信息构建会话工厂 SqlSessionFactory。

）创建会话对象：由会话工厂创建 SqlSession 对象，该对象中包含了执行 SQL 语句的所有方法。

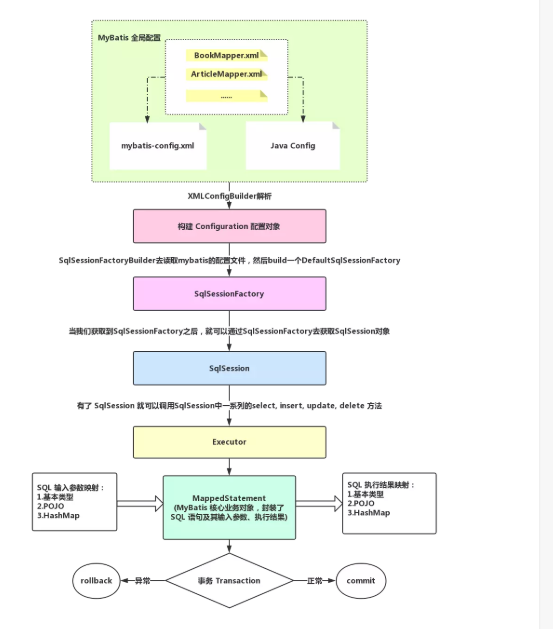
5）Executor 执行器：MyBatis 底层定义了一个 Executor 接口来操作数据库，它将根据 SqlSession 传递的参数动态地生成需要执行的 SQL 语句，同时负责查询缓存的维护。

6）MappedStatement 对象：在 Executor 接口的执行方法中有一个 MappedStatement 类型的参数，该参数是对映射信息的封装，用于存储要映射的 SQL 语句的 id、参数等信息。

7）输入参数映射：输入参数类型可以是 Map、List 等集合类型，也可以是基本数据类型和 POJO 类型。输入参数映射过程类似于 JDBC 对 preparedStatement 对象设置参数的过程。

8）输出结果映射：输出结果类型可以是 Map、 List 等集合类型，也可以是基本数据类型和 POJO 类型。输出结果映射过程类似于 JDBC 对结果集的解析过程。

MyBatis 框架的执行流程图:



### 2.2.4 Spring boot整合Redis

**1.redis简介**

redis是一个key-value。和Memcached类似，它支持存储的value类型相对更多，包括string(字符串)、list(链表)、set(集合)、zset(sorted set --有序集合)和hash（哈希类型）。这些数据类型都支持push/pop、add/remove及取交集并集和差集及更丰富的操作，而且这些操作都是原子性的。在此基础上，redis支持各种不同方式的排序。与memcached一样，为了保证效率，数据都是缓存在内存中。区别的是redis会周期性的把更新的数据写入磁盘或者把修改操作写入追加的记录文件，并且在此基础上实现了master-slave(主从)同步。Redis 是一个开源（BSD许可）的，内存中的数据结构存储系统，它可以用作数据库、缓存和消息中间件。 它支持多种类型的数据结构，如 字符串（strings）， 散列（hashes）， 列表（lists）， 集合（sets）， 有序集合（sorted sets）与范围查询， bitmaps， hyperloglogs和 地理空间（geospatial）索引半径查询。 Redis 内置了 复制（replication），LUA脚本（Lua scripting）， LRU驱动事件（LRU eviction），事务（transactions）和不同级别的 磁盘持久化（persistence）， 并通过 Redis哨兵（Sentinel）和自动 分区（Cluster）提供高可用性（high availability）。

**2.Redis特点**

Redis将其数据库完全保存在内存中，仅使用磁盘进行持久化。

与其它键值数据存储相比，Redis有一组相对丰富的数据类型。

Redis可以将数据复制到任意数量的从机中。

**3.Redis优点**

异常快 - Redis非常快，每秒可执行大约110000次的设置(SET)操作，每秒大约可执行81000次的读取/获取(GET)操作。

支持丰富的数据类型 - Redis支持开发人员常用的大多数数据类型，例如列表，集合，排序集和散列等等。这使得Redis很容易被用来解决各种问题，因为我们知道哪些问题可以更好使用地哪些数据类型来处理解决。

操作具有原子性 - 所有Redis操作都是原子操作，这确保如果两个客户端并发访问，Redis服务器能接收更新的值。

多实用工具 - Redis是一个多实用工具，可用于多种用例，如：缓存，消息队列(Redis本地支持发布/订阅)，应用程序中的任何短期数据，例如，web应用程序中的会话，网页命中计数等。

## 2.3 实习内容

### 2.3.1 阅读相关文献

阅读与Springboot 相关的论文与资料，学习什么是Springboot、Springboot的作用,以及与其他各种技术是如何整合的；以及java的各种类包

相关文献包含但不限于：

<http://springboot.fun/> 此网站中包含了有关springboot大量的学习内容，还有一些springboot实例供我们学习。

《Spring boot实战》 本书以Spring应用程序开发为中心，全面讲解如何运用Spring Boot提高效率，使应用程序的开发和管理更加轻松有趣。作者行文亲切流畅，以大量示例讲解了Spring Boot在各类情境中的应用，内容涵盖起步依赖、Spring Boot CLI、Groovy、Grails、Actuator。

《Spring Boot+Vue全栈开发实战》

《Spring boot编程思想》 本书是《Spring Boot 编程思想》的核心篇，开篇总览Spring Boot核心特性，接着讨论自动装配（Auto-Configuration）与SpringApplication。《Spring Boot编程思想（核心篇）》的讨论以Spring Boot为中心，议题发散至Spring技术栈、JSR及Java。希望透过全局的视角，帮助读者了解Spring Boot变迁的历程；经过多方的比较，帮助读者理解Spring Boot特性的原理；整合标准的规范，帮助读者掌握Spring Boot设计的哲学

《Springboot揭秘》本书以微服务的基本概念介绍性开篇，逐步引出Java平台下打造微服务的利器SpringBoot微框架。书中从SpringBoot微框架的“出身”开始，循序渐进，为大家剖析SpringBoot微框架的设计理念和原理，并对框架的重点功能和模块进行了逐一讲解。

《JavaEE开发的颠覆者: Spring Boot实战》 在当今Java EE 开发中，Spring 框架是当之无愧的王者。而Spring Boot 是Spring 主推的基于“习惯优于配置”的原则，让你能够快速搭建应用的框架，从而使得Java EE 开发变得异常简单。

《JavaEE开发的颠覆者: Spring Boot实战》从Spring 基础、Spring MVC 基础讲起，从而无难度地引入Spring Boot 的学习。涵盖使用Spring Boot 进行Java EE 开发的绝大数应用场景，包含：Web 开发、数据访问、安全控制、批处理、异步消息、系统集成、开发与部署、应用监控、分布式系统开发等。

### 2.3.2 JAVA环境搭建以及IDEA下载

在64位windows上安装jdk1.8.0\_202,并且配置相应的环境变量。

1. 环境变量配置

1.1 下载jdk1.8.0\_202

<https://www.oracle.com/technetwork/java/javase/downloads/index.html>

1.2 JAVA\_HOME环境变量设置

1）首先右击【计算机】进入【属性】然后选择其中的【高级系统设置】。

2）点击进入【高级】中的【环境变量】，进入环境变量编辑界面.

3）在下方的【系统变量】中，并不存在JAVA\_HOME变量,那么我们需要点击【新建】

4）变量名输入：JAVA\_HOME

变量值指的是实际的安装路径（比如小编的路径为：C:\Program Files\Java\jdk-9）。

最后点击确定

5）同样是在【系统变量中】我们可以看到path变量已经存在，那么我们只需要点击【编辑】，进入path变量的编辑

6）变量名不变，变量值改为【%JAVA\_HOME%\bin;%JAVA\_HOME%\jre\bin;】，最后点击确定

7）测试

通过CMD命令Java –version

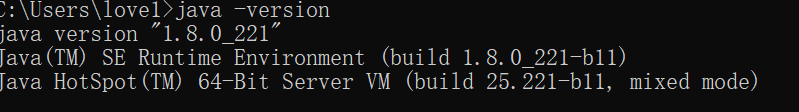


图2.11

1. 安装IDEA
   1. 下载IDEA

<https://www.jetbrains.com/idea/download/#section=windows>

1. 部署一个Springboot项目

3.1 在IDEA中新建一个springboot web项目

3.2 在pom文件中引入相关依赖（引入依赖的目的是引入相关的jar包，此jar包由maven自动引入）

<dependencies>

<!-- <dependency>-->

<!-- <groupId>org.springframework.boot</groupId>-->

<!-- <artifactId>spring-boot-starter-security</artifactId>-->

<!-- </dependency>-->

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-thymeleaf</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-data-redis</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web-services</artifactId>

</dependency>

<!-- <dependency>-->

<!-- <groupId>org.springframework.session</groupId>-->

<!-- <artifactId>spring-session-core</artifactId>-->

<!-- </dependency>-->

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

<!-- <dependency>-->

<!-- <groupId>org.springframework.security</groupId>-->

<!-- <artifactId>spring-security-test</artifactId>-->

<!-- <scope>test</scope>-->

<!-- </dependency>-->

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

</plugins>

</build>

1. 测试web

在控制类中书写以下代码

@ResponseBody

@RequestMapping("/index1")

public String index1(){

return "hello world";

}

然后再浏览器中输入<http://localhost:8080/index1> 会在界面输出 hello world 说明部署成功。

### 2.3.3 Springboot web代码学习

spring-boot 非常适合开发web应用的

静态资源访问：

在我们开发Web应用的时候，需要引用大量的js、css、图片等静态资源。

默认配置

Spring Boot默认提供静态资源目录位置需置于classpath下，目录名需符合如下规则：

/static

/public

/resources

举例：我们可以在src/main/resources/目录下创建static，在该位置放置一个图片文件。启动程序后，尝试访问http://localhost:8080/D.jpg。如能显示图片，配置成功。

渲染Web页面：

在之前的示例中，我们都是通过@RestController来处理请求，所以返回的内容为json对象。那么如果需要渲染html页面的时候，要如何实现呢？

模板引擎

在动态HTML实现上Spring Boot依然可以完美胜任，并且提供了多种模板引擎的默认配置支持，所以在推荐的模板引擎下，我们可以很快的上手开发动态网站。

Spring Boot提供了默认配置的模板引擎主要有以下几种：

Thymeleaf

FreeMarker

Velocity

Groovy

Mustache

Spring Boot建议使用这些模板引擎，避免使用JSP，若一定要使用JSP将无法实现Spring Boot的多种特性，具体可见后文：支持JSP的配置

当你使用上述模板引擎中的任何一个，它们默认的模板配置路径为：src/main/resources/templates。当然也可以修改这个路径，具体如何修改，可在后续各模板引擎的配置属性中查询并修改。

Thymeleaf

Thymeleaf是一个XML/XHTML/HTML5模板引擎，可用于Web与非Web环境中的应用开发。它是一个开源的Java库，基于Apache License 2.0许可，由Daniel Fernández创建，该作者还是Java加密库Jasypt的作者。

Thymeleaf提供了一个用于整合Spring MVC的可选模块，在应用开发中，你可以使用Thymeleaf来完全代替JSP或其他模板引擎，如Velocity、FreeMarker等。Thymeleaf的主要目标在于提供一种可被浏览器正确显示的、格式良好的模板创建方式，因此也可以用作静态建模。你可以使用它创建经过验证的XML与HTML模板。相对于编写逻辑或代码，开发者只需将标签属性添加到模板中即可。接下来，这些标签属性就会在DOM（文档对象模型）上执行预先制定好的逻辑。

#### springboot 注解

Springboot中的注解是非常重要的，注解大量简化了配置文件，使得很多配置信息，不再需要配置文件，只需要使用注解就可以实现，大量减少了编程的负担。

@SpringBootConfiguration 继承至@Configuration，对于熟悉spring的开发者而言，此标注当前类是配置类，并会将当前类内声明的一个或多个以@Bean注解标记的方法的实例纳入到srping容器中，并且实例名就是方法名。

@EnableAutoConfiguration 这个注解就是springboot能自动进行配置的魔法所在了。主要是通过此注解，能所有符合自动配置条件的bean的定义加载到spring容器中，比如根据spring-boot-starter-web ，来判断你的项目是否需要添加了webmvc和tomcat，就会自动的帮你配置web项目中所需要的默认配置。具体的使用，会在后期自定义实现一个自动启动类时，会讲解到它的一些机制。此章节就不深入了，只需要它是这个用途即可，一般上也单独使用不要这个注解，但比如需要排除一些无需自动配置的类时，可利用exclude进行排除。

@ComponentScan 这个熟悉spring的开发者也应该熟悉，会扫描当前包及其子包下被@Component，@Controller，@Service，@Repository等注解标记的类并纳入到spring容器中进行管理。

@Controller 和 @RestController

@RestController 是Spring4之后加入的注解，原来在@Controller中返回json需要@ResponseBody来配合，如果直接用@RestController替代@Controller就不需要再配置@ResponseBody，默认返回json格式。而@Controller是用来创建处理http请求的对象，一般结合@RequestMapping使用。

@RequestMapping一个用来处理请求地址映射的注解，可用于类或方法上。用于类上，表示类中的所有响应请求的方法都是以该地址作为父路径。

常用属性（摘抄至网络）：

value， method；

value： 指定请求的实际地址，指定的地址可以是URI Template 模式（后面将会说明）；

method： 指定请求的method类型， GET、POST、PUT、DELETE等；

consumes，produces；

consumes： 指定处理请求的提交内容类型（Content-Type），例如application/json, text/html;

produces: 指定返回的内容类型，仅当request请求头中的(Accept)类型中包含该指定类型才返回；

params，headers；

params： 指定request中必须包含某些参数值是，才让该方法处理。

headers： 指定request中必须包含某些指定的header值，才能让该方法处理请求。

常用的基本上就value和method了。

其简化注解有

@GetMapping 等同于 @RequestMapping(method = RequestMethod.GET)

@PostMapping 等同于 @RequestMapping(method = RequestMethod.POST)

@PutMapping 等同于 @RequestMapping(method = RequestMethod.PUT)

@DeleteMapping 等同于 @RequestMapping(method = RequestMethod.DELETE)

@PatchMapping 等同于 @RequestMapping(method = RequestMethod.PATCH)

@RequestBody和@ResponseBody

@RequestBody注解允许request的参数在reqeust体中，常常结合前端POST请求，进行前后端交互。@ResponseBody注解支持将的参数在reqeust体中，通常返回json格式给前端。

@PathVariable、@RequestParam、@RequestAttribute

@PathVariable用来接收参数,如/path/001,可接收001作为参数

@RequestParam 用来接收URL中的参数,如/param?id=001,可接收001作为参数

@Component、@Service、@Repository

这三者都是申明一个单例的bean类并纳入spring容器中，后两者其实都是继承于@Component。@Component 最普通的组件，可以被注入到spring容器进行管理@Repository 作用于持久层@Service 作用于业务逻辑层

#### web前后端交互

web的前后端交互大多数都是用AJAX和表单实现的

1.通过form表单以post/get方式提交数据。

当你点击submit按钮时，浏览器会默认把你在input里面输入的数据，以post或get的方式提交到form表单中的action这个地址。相当于你提交表单时，就会向服务器发送一个请求，然后服务器会接受并处理提交过来的form表单，最后返回一个新的网页。你可以结合以下代码来理解这段话。

<form action="/form.html" method="post/get">

<input type="text" name="username" placeholder="username">

<input type="password" name="password" placeholder="password">

<input type="submit">

</form>

【补充】

1、get提交数据：请求参数（一般为input里的值）拼装成url，相当于向服务器发url请求。

2、post提交数据：直接向服务器发请求，参数直接发给后台。

但是，这种方法会导致几个问题：

1、在提交时，页面会发生跳转或刷新，导致用户体验不好。

2、单项提交，把数据提交给后台，但是不知道后台会给出怎样的响应，因为提交后页面就发生跳转了。比如：用户登录，那么就不知道到底是注册成功了还是失败了。

3、浪费宽带。因为前后两个页面中的大部分HTML代码往往是相同的。但由于每次应用的交互都需要向服务器发送请求，应用的响应时间就依赖于服务器的响应时间，这就导致了用户界面的响应比本地应用慢的多。

2.通过ajax进行前后端交互

Ajax简介：

AJAX = 异步 JavaScript 和 XML。

　AJAX 是一种用于创建快速动态网页的技术。

　AJAX 是一种借助XML，HTML，CSS和Java Script创建更好，更快，更交互的Web应用程序的新技术。

　AJAX 是一种独立于Web服务器软件的Web浏览器技术。

　AJAX 使用XHTML作为内容，CSS使用文档对象模型和JavaScript进行动态内容显示。

　通过在后台与服务器进行少量数据交换，AJAX 可以使网页实现异步更新。这意味着可以在不重新加载整个网页的情况下，对网页的某部分进行更新。

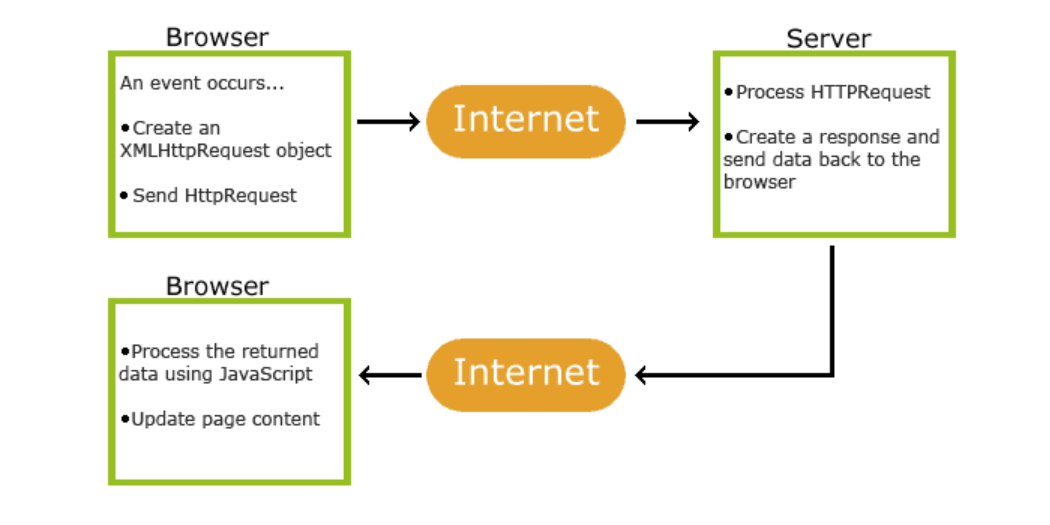
　传统的网页（不使用 AJAX）如果需要更新内容，必需重载整个网页面。

　使用AJAX，当您点击提交时，JavaScript将向服务器发出请求，解释结果并更新当前屏幕。

　当客户端程序在后台请求来自服务器的信息时，用户可以继续使用该应用程序。

　数据驱动而不是页面驱动。

AJAX工作原理如下图所示：



AJAX优缺点

(1).AJAX的优点

<1>.无刷新更新数据。AJAX最大优点就是能在不刷新整个页面的前提下与服务器通信维护数据。这使得Web应用程序更为迅捷地响应用户交互，并避免了在网络上发送那些没有改变的信息，减少用户等待时间，带来非常好的用户体验。

<2>.异步与服务器通信。

AJAX使用异步方式与服务器通信，不需要打断用户的操作，具有更加迅速的响应能力。优化了Browser和Server之间的沟通，减少不必要的数据传输、时间及降低网络上数据流量。

<3>.前端和后端负载平衡。

AJAX可以把以前一些服务器负担的工作转嫁到客户端，利用客户端闲置的能力来处理，减轻服务器和带宽的负担，节约空间和宽带租用成本。并且减轻服务器的负担，AJAX的原则是“按需取数据”，可以最大程度的减少冗余请求和响应对服务器造成的负担，提升站点性能

<4>.基于标准被广泛支持。

AJAX基于标准化的并被广泛支持的技术，不需要下载浏览器插件或者小程序，但需要客户允许JavaScript在浏览器上执行。随着Ajax的成熟，一些简化Ajax使用方法的程序库也相继问世。同样，也出现了另一种辅助程序设计的技术，为那些不支持JavaScript的用户提供替代功能。

<5>.界面与应用分离。

Ajax使WEB中的界面与应用分离（也可以说是数据与呈现分离），有利于分工合作、减少非技术人员对页面的修改造成的WEB应用程序错误、提高效率、也更加适用于现在的发布系统。

(2).AJAX的缺点

<1>.AJAX干掉了Back和History功能，即对浏览器机制的破坏。

在动态更新页面的情况下，用户无法回到前一个页面状态，因为浏览器仅能记忆历史记录中的静态页面。一个被完整读入的页面与一个已经被动态修改过的页面之间的差别非常微妙；用户通常会希望单击后退按钮能够取消他们的前一次操作，但是在Ajax应用程序中，这将无法实现。

<2>.AJAX的安全问题。

AJAX技术给用户带来很好的用户体验的同时也对IT企业带来了新的安全威胁，Ajax技术就如同对企业数据建立了一个直接通道。这使得开发者在不经意间会暴露比以前更多的数据和服务器逻辑。Ajax的逻辑可以对客户端的安全扫描技术隐藏起来，允许黑客从远端服务器上建立新的攻击。还有Ajax也难以避免一些已知的安全弱点，诸如跨站点脚步攻击、SQL注入攻击和基于Credentials的安全漏洞等等。

<3>.对搜索引擎支持较弱。

对搜索引擎的支持比较弱。如果使用不当，AJAX会增大网络数据的流量，从而降低整个系统的性能。

<4>.破坏程序的异常处理机制。

<5>.违背URL和资源定位的初衷。

例如，我给你一个URL地址，如果采用了Ajax技术，也许你在该URL地址下面看到的和我在这个URL地址下看到的内容是不同的。这个和资源定位的初衷是相背离的。

<6>.AJAX不能很好支持移动设备。

#### 微信支付

我们做的微信支付使用的是native支付方式

#### 部分核心函数解析

从web项目中，摘选出该web项目的一些核心类以及方法的源代码实现。

**加密算法类AES加密算法和MD5加密**

/\*\*\*

\* MD5加盐加密

\* @param string

\* @param salt

\* @return

\*/

public static String md5Encryption(String string, String salt) {

return getUpperCaseMd5(string + salt);

}

/\*\*\*

\* 加密后英文是大写

\* @param src

\* @return

\*/

private static String getUpperCaseMd5(String src) {

try {

StringBuffer stringBuffer = new StringBuffer();

MessageDigest messageDigest = MessageDigest.getInstance(MESSAGE\_DIGEST\_ALGORITHM);

byte[] data = messageDigest.digest(src.getBytes());

//遍历字节数组，取出高四位和低四位对应的字符,最终会得到长度为32的字符串

for (byte b : data) {

//高4位

stringBuffer.append(CHARS[(b >> 4) & HEX\_HELPER]);

//低4位

stringBuffer.append(CHARS[b & HEX\_HELPER]);

}

return stringBuffer.toString();

} catch (Exception e) {

// TODO: handle exception

log.info("[大写MD5生成error]");

}

return null;

}

/\*\*\*

\* AES加密

\* @param password

\* @param key

\* @return

\* @throws Exception

\*/

private static String encryptHelper(String password, String key) throws Exception {

//创建密码器

Cipher cipher = Cipher.getInstance(DEFAULT\_CIPHER\_ALGORITHM);

byte[] bytes = password.getBytes(DEFAULT\_CHARSET);

//初始化密码器

cipher.init(Cipher.ENCRYPT\_MODE, getSecretKeySpec(key));

//加密

byte[] bytes1 = cipher.doFinal(bytes);

//通过BASE64转码返回

return Base64.encodeBase64String(bytes1);

}

/\*\*\*

\* AES解密

\* @param password

\* @param key

\* @return

\* @throws Exception

\*/

private static String decryptHelper(String password, String key) throws Exception {

Cipher cipher = Cipher.getInstance(DEFAULT\_CIPHER\_ALGORITHM);

//使用密钥初始化解密

cipher.init(Cipher.DECRYPT\_MODE, getSecretKeySpec(key));

byte[] bytes = cipher.doFinal(Base64.decodeBase64(password));

return new String(bytes, DEFAULT\_CHARSET);

}

/\*\*\*

\* 生成加密密钥

\* @param password

\* @return

\* @throws NoSuchAlgorithmException

\*/

private static SecretKeySpec getSecretKeySpec(final String password)

throws NoSuchAlgorithmException {

//返回密钥生成器对象

KeyGenerator keyGenerator = KeyGenerator.getInstance(KEY\_ALGORITHM);

SecureRandom secureRandom = SecureRandom.getInstance(PRNG\_ALGORITHM);

secureRandom.setSeed(password.getBytes());

//设置AES密钥长度

keyGenerator.init(SECRET\_KEY\_LENGTH, secureRandom);

//生成一个密钥

SecretKey secretKey = keyGenerator.generateKey();

//转换为AES密钥

return new SecretKeySpec(secretKey.getEncoded(), KEY\_ALGORITHM);

}

**ftp上传和删除文件**

/\*\*

\* 上传文件

\*

\* @param file

\* @param path

\* @return

\* @throws IOException

\*/

public String uploadFile(MultipartFile file, String path) throws IOException {

FTPClient ftpClient = ftpPool.getFtpClient();

ftpClient.changeWorkingDirectory(path);

String fileName = UUIDUtil.getUuid() + file.getOriginalFilename()

.substring(file.getOriginalFilename().lastIndexOf("."));

try (InputStream inputStream = file.getInputStream()) {

//执行文件传输

boolean result = ftpClient.storeFile(fileName, inputStream);

if (!result) {

//上传失败

log.info("上传文件失败");

}

} catch (Exception e) {

log.warn("上传文件异常", e);

return null;

} finally {

//关闭输入流

log.info("开始归还FTP连接");

//归还连接

ftpPool.returnFtpClient(ftpClient);

}

return fileName;

}

/\*\*

\* 删除FTP上的文件

\*

\* @param fileName

\* @param path

\*/

public boolean deleteFile(String fileName, String path) throws IOException {

FTPClient ftpClient = ftpPool.getFtpClient();

ftpClient.changeWorkingDirectory(path);

boolean deleted = false;

try {

deleted = ftpClient.deleteFile(fileName);

} catch (IOException e) {

log.warn("删除文件异常:{}", e);

} finally {

//关闭输入流

log.info("开始归还FTP连接");

//归还连接

ftpPool.returnFtpClient(ftpClient);

}

return deleted;

}

public FtpConfig getFtpConfig() {

return ftpConfig;

}

public FtpPool getFtpPool() {

return ftpPool;

}

**session的运用**

/\*\*

\* 获取session

\*

\* @param sessionKey

\* @return

\* @throws UnknownSessionException

\*/

@Override

protected Session retrieveSession(SessionKey sessionKey) throws UnknownSessionException {

//获取sessionId

Serializable sessionId = getSessionId(sessionKey);

ServletRequest request = null;

if (sessionKey instanceof WebSessionKey) {

request = ((WebSessionKey) sessionKey).getServletRequest();

}

if (request != null && sessionId != null) {

//从request中获取session

Session session = (Session) request.getAttribute(sessionId.toString());

if (session != null) {

return session;

}

}

Session session = super.retrieveSession(sessionKey);

if (request != null && sessionId != null) {

//将从redis中读取的session放入request中

request.setAttribute(sessionId.toString(), session);

}

return session;

## 2.4 实习工作总结

虽然这次实习项目较新、涉及到非常多没接触过的知识，也遇到了许多困难，但最终逐一解决了问题后，我本人也收获了很多。以下是我对本次实习工作的总结与体会。

1. 阅读论文及相关资料。

首先是java编程语言的学习，通过一段时间的学习与代码，对Java有了一定的了解，对web项目开发有了很大的帮助。

1. 环境搭建

Web开发是在Java 8环境下进行开发的。

1. 学习代码

首次运用Java以及Springboot框架开发web，肯定会很陌生，最好的办法就是找一些web项目实例，看一些别人写的代码，对初学者有着很大的帮助。

经过3周的实习，我个人受益匪浅。对web设计也有了很深的了解。Web分前后端，以及前后端交互，web的框架，以及web框架与数据库和缓存的整合。也了解了有关web账户安全的问题。对微信支付也有了一定的了解，微信支付需要调用各种微信支付的API，也需要准确的传送各种有效的参数，以及对微信支付的返回值做精确的解析，还有微信支付的各种加密算法，安全问题等。

## 2.5 实习建议

1. 实习期间应多与他人交流，每个人的实习经验都非常宝贵，应相互交流、学习，多问，多学，多去实践。

2. 主动性要强，大多数人都在忙着自己的事，主动去分担一些工作，主动学习一些相关的知识。

3. 实习是一次很好的学习和掌握当前前沿技术的机会，一定要认真对待。

4. 如果打算毕业直接工作的，应注重锻炼自己的工程能力，增加一些相关方面的经验。

5. 基础知识的学习一定不能拉下来。

# 3 实习问题解答

问题：微信支付的具体流程

回答： 微信支付为了保证交易的安全性，一律采用的时https传输，采用的时post提交方法，每一个接口所要提交的数据以及返回的数据的格式都是XML格式，需要自己对字段进行xml解析。字符编码的格式为UTF-8字符编码。签名算法采用的是MD5和HMAC-SHA256，在请求和接收数据的时候均需要校验签名，调用调用申请退款、撤销订单、红包接口等需要商户api证书。微信支付的第一步是统一下单，需要调用的接口链接是

URL地址：https://api.mch.weixin.qq.com/pay/unifiedorder

URL地址：https://api2.mch.weixin.qq.com/pay/unifiedorder(备用域名)见跨城冗灾方案，统一下单是不需要证书的，必须请求的字段appid, mac\_id, nonce\_str, sign, body, out\_trade\_no, total\_fee, spbill\_create\_ip。需要将这些字段已经对应的数据都变成XML格式，才能提交发送。然后根据返回的结果判断订单交易的结果，如果只有return\_code返回为SUCCESS，说明此次订单失败，会返回错误代码err\_code\_des, 只有当return\_code 和result\_code都为SUCCESS的时候订单成功，并且返回一个重要的信息即二维码的链接，由此链接可以在前端生成订单的二维码，扫码即可完成支付。在微信支付中申请退款也是比较重要的，申请退款发生在当交易发生之后一段时间内，由于买家或者卖家的原因需要退款时，卖家可以通过退款接口将支付款退还给买家，微信支付将在收到退款请求并且验证成功之后，按照退款规则将支付款按原路退到买家帐号上。注意：

1、交易时间超过一年的订单无法提交退款

2、微信支付退款支持单笔交易分多次退款，多次退款需要提交原支付订单的商户订单号和设置不同的退款单号。申请退款总金额不能超过订单金额。 一笔退款失败后重新提交，请不要更换退款单号，请使用原商户退款单号

3、请求频率限制：150qps，即每秒钟正常的申请退款请求次数不超过150次 错误或无效请求频率限制：6qps，即每秒钟异常或错误的退款申请请求不超过6次

4、每个支付订单的部分退款次数不能超过50次

申请退款的接口链接为<https://api.mch.weixin.qq.com/secapi/pay/refund>，申请退款是需要双向证书的。

在支付=完成之后，需要明确知道自己是否支付成功，这时就需要用到支付结果通知这个API ,该链接是通过统一下单API中的提交的参数notify\_nul中设置的，只有有了这个字段，才会收到微信通知，通知url必须为直接可访问的url，不能携带参数，该API市不需要证书的。主要应用的场景是在支付支付完成后，微信会把相关支付结果及用户信息通过数据流的形式发送给商户，商户需要接收处理，并按文档规范返回应答。1、同样的通知可能会多次发送给商户系统。商户系统必须能够正确处理重复的通知。2、后台通知交互时，如果微信收到商户的应答不符合规范或超时，微信会判定本次通知失败，重新发送通知，直到成功为止（在通知一直不成功的情况下，微信总共会发起多次通知，通知频率为15s/15s/30s/3m/10m/20m/30m/30m/30m/60m/3h/3h/3h/6h/6h - 总计 24h4m），但微信不保证通知最终一定能成功。3、在订单状态不明或者没有收到微信支付结果通知的情况下，建议商户主动调用微信支付【查询订单API】确认订单状。需要特别注意的是商户系统对于支付结果通知的内容一定要做签名验证,并校验返回的订单金额是否与商户侧的订单金额一致，防止数据泄漏导致出现“假通知”，造成资金损失。退款结果通知与支付结果通知大致类似，也需要在申请退款的接口中上传参数notify\_url以开通该功能，并且通知url必须为直接可访问的url，不能携带参数。应用场景是当商户申请的退款有结果后，微信会把相关结果发送给商户，商户需要接收处理，并返回应答。

对后台通知交互时，如果微信收到商户的应答不是成功或超时，微信认为通知失败，微信会通过一定的策略定期重新发起通知，尽可能提高通知的成功率，但微信不保证通知最终能成功（通知频率为15s /15s /30s /3m /10m /20m /30m /30m /30m/60m/3h/3h/3h/6h/6h - 总计 24h4m）。注意：同样的通知可能会多次发送给商户系统。商户系统必须能够正确处理重复的通知。推荐的做法是，当收到通知进行处理时，首先检查对应业务数据的状态，判断该通知是否已经处理过，如果没有处理过再进行处理，如果处理过直接返回结果成功。在对业务数据进行状态检查和处理之前，要采用数据锁进行并发控制，以避免函数重入造成的数据混乱。

特别说明：退款结果对重要的数据进行了加密，商户需要用商户秘钥进行解密后才能获得结果通知的内容。解密步骤如下：（1）对加密串A做base64解码，得到加密串B（2）对商户key做md5，得到32位小写key\* ( key设置路径：微信商户平台(pay.weixin.qq.com)-->账户设置-->API安全-->密钥设置 )

（3）用key\*对加密串B做AES-256-ECB解密（PKCS7Padding）。

# 4 生产实习指标点以及问题回答

**1.设计/开发解决方案**

模块化的设计以及设计模式的产生，对软件工业的设计雄安率有了很大的提升，模块化的设计使得设计可以分工合作，设计模式会使得代码的利用率大大提升，使得代码能更好的维护和开发。还有在实习过程中使用了大量的注解，注解的产生使得在软件开发过程中可以省略配置文件的开发，使得开发事倍功半，也使得代码更加简化易懂。

**2.工程与社会**

可以使用计算机和互联网给传统的行业模式，提供一种新型的模式，通常我们做实验就会去实验室，可以通过计算机和网络在计算机上虚拟化出来一种实验的环境，可以在网络上进行实验活动，方便快捷。

**3.环境和可持续发展**

有些传统的行业是有污染的，且不可持续发展，就可以用计算机和互联网进行解决。虚拟化实验室可以解决实验器材和实验药品的问题，可以不用再去购买器材和药品，也避免了实验中所造成的化学污染，以及对人可能造成的伤害。

**4.职业规范**

1.对自己的工作承担完全的责任；2.对所从事的工作必须确认是安全的，符合法律的，符合规格说明的，经过合格测试的，不会降低生活品质的，不破化环境的，不会对他人利益造成伤害的；3当他们所从事的设计工程对用户，公众或者环境造成任何实际或潜在的危害时，应立即向适当的人或者有关当局揭露举报；4 应将致力于将自己的专业技能应用于公众事业和公共教育的发展。

**5.个人或团队**

当今社会关系离不开团结协作，任何行业也都离不开团结协作，具备团队意识和团队协作的能力是任何人都必须具备的能力，能够在多学科背景下的团队中协同工作，并承担个体、团队成员以及负责人的角色。在实习过程中我们也有一个四人的团队，其中一个人是队长，负责整个项目的统筹规划和任务的分配以及项目中大半的工作，而我则是其中一位队员，负责接口文档的书写以及测试和微信支付模块的开发

**6.沟通**

任何科学的进步，科技的传播都离不开有效的沟通，我们在科学研究过程中业界同行及社会公众进行有效沟通和交流。我们也需要时刻关注行业动态，及时掌握最新的科技，跟上时代的步伐，我们在科学研究的过程中需要撰写报告和设计文稿、陈述发言、清晰表达或回应指令，并具备一定的国际视野，能够在跨文化背景下进行沟通和交流。我们也需要具备一定的阅读外国文献的能力。

当前的软件开发都是在一定的框架中完成的，而且需要懂得各种技术，微服务，微架构，缓存，持久化，分布式等各种技术

**7.项目管理**

工程项目管理需要运用各种知识，手段，技能和方法实现预定的项目目标。工程项目管理的知识，技能，手段和方法很多，并且不断发展，但工程项目管理的基本原理主要是系统管理和过程管理。工程管理一般先决定整体框架，后进入详细的设计程序，强调系统整体功能最佳。项目管理主要有以下几种方法：阶段化管理，量化管理，优化管理等；我们主要使用了阶段化管理和优化管理管理方法。阶段化管理分为项目的准备和启动，项目计划，项目实施，项目验收和项目总结等，阶段化的管理方法，使得一个项目的过程清晰可见，结构完备，使得开发更加方便效率更高

**8.终身学习**

当今世界，科技发展日新月异，已经没有一种技术是可以永不过时的，一种技术被另一种技术替代是一种常态，我们已经不能满足于学一门技术而受益于终身，这种学习方式已经不能适合当今世界的发展，当今世界需要我们时刻不断学习，才能在不断竞争中脱颖而出；故此我们需要养成一种终生学习的好习惯，才能应对接下来的全球化的竞争，我们应当和技术，知识，科技同步的发展才能使我们立于不败之地。

。

# 参考文献

[1]https://www.cnblogs.com/zhoudi/p/5820513.html

[2]https://blog.csdn.net/Chen\_Victor/article/details/81262233

[3]https://blog.csdn.net/qq\_39470733/article/details/80980320

[4]https://www.jianshu.com/p/83f77a937c05

[5]https://spring.io/projects/spring-boot

[6]https://spring.io/projects/spring-security

[7]https://spring.io/projects/spring-session

[8]https://spring.io/projects/spring-ws

[9]https://www.oracle.com/technetwork/java/javase/downloads/index.html