****

****

毕业论文

基于SpringBoot和Vue的仓库管理系统

设计与实现

|  |  |
| --- | --- |
| 院 别 | 数学与统计学院 |
| 专业名称 | 数据科学与大数据技术 |
| 班级学号 | **1901-20199465** |
| 学生姓名 | 罗悦桐 |
| 指导教师 | 王子健 实验师 |

**2023年5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： 日期：

基于SpringBoot和Vue的仓库管理系统设计与实现

摘 要

本论文旨在探究一种基于SpringBoot和Vue的仓库管理系统前后端分离设计与实现的方案。该系统综合应用了多种技术，包括前端方面的HTML、CSS、JavaScript（ES6）、Vue.JS、ElementUI、Node.JS，以及后端方面的Java、SpringBoot、Maven、MyBatis、MYSQL等。系统实现的功能涵盖管理员管理、用户管理、仓库管理、记录管理等方面。

在绪论部分，本文将介绍当前企业信息化转型趋势以及对于仓库管理系统的具体需求，阐述前后端分离开发的优势和该架构模式对于提高系统的可维护性和可拓展性的作用。在论文主体部分，将详细介绍该系统的设计和实现方案，涵盖页面布局的拆分、组件化开发、增删改查功能的实现、权限管理等方面。其中，权限管理可实现区分关联本地MYSQL数据库的不同权限用户的系统功能，ElementUI的丰富组件也有助于提高系统的用户实际体验。

本文还将介绍基于B/S架构的Web应用程序开发模式，包括前端发起请求后，SpringBoot的MVC架构（控制层、业务逻辑层和数据访问层）逐层返回结果给用户的流程原理。

通过本论文的研究，不仅能够了解前后端分离的架构模式，还能够掌握JavaWeb的开发技术，为企业实现信息化转型提供一定的帮助。

关键词：仓库管理系统；前后端分离；B/S架构；SpringBoot；Vue

**Design and Implementation of Warehouse Management System Based on Spring Boot and Vue**

**Abstract**

This paper aims to explore a solution for front-end and back-end separation design and implementation of a warehouse management system based on SpringBoot and Vue. The system integrates multiple technologies, including HTML, CSS, JavaScript (ES6), Vue.JS, ElementUI, Node.JS on the front-end side, and Java, SpringBoot, Maven, MyBatis, MYSQL on the back-end side. The functions implemented by the system cover aspects such as administrator management, user management, warehouse management and logistics management.

In the introduction section, this thesis introduces the current trend of enterprise informationization transformation and the specific requirements for warehouse management systems. It also explores the advantages of front-end and back-end separation development and discusses the role of this architectural pattern in improving the system's maintainability and scalability. In the main body of the paper, the design and implementation schemes of the system will be described in detail, covering aspects such as page layout separation, component development, implementation of CRUD functions, and permission management. The permission management can realize the system function of distinguishing different permission users associated with the local MYSQL database, and the rich components of ElementUI can also help improve the user experience of the system.

This thesis will also introduce the web application development mode based on B/S architecture, including the principle of SpringBoot's MVC architecture (presentation layer, business logic layer, and data access layer) returning results to users layer by layer after the front-end initiates a request.

Through the research in this thesis, not only can we understand the architecture pattern of front-end and back-end separation, but we can also master the development technology of JavaWeb, providing certain help for enterprises to realize informationization transformation.

**Key Words**: Warehouse management system; Front-end and back-end separation; B/S architecture; SpringBoot; Vue

目 录

[1 绪论 1](#_Toc137306327)

[1.1 研究背景与意义 1](#_Toc137306328)

[1.2 研究内容和贡献 2](#_Toc137306329)

[1.3 论文组织结构 3](#_Toc137306330)

[2 关键技术介绍 4](#_Toc137306331)

[2.1 后端开发技术 4](#_Toc137306332)

[2.1.1 Java语言 4](#_Toc137306333)

[2.1.2 SpringBoot 4](#_Toc137306334)

[2.1.3 Maven 7](#_Toc137306335)

[2.1.4 Mybatis 8](#_Toc137306336)

[2.2 前端开发技术 9](#_Toc137306337)

[2.2.1 H5C3 9](#_Toc137306338)

[2.2.2 Vue.js 9](#_Toc137306339)

[2.3 B/S架构模式 10](#_Toc137306340)

[2.4 前后端分离架构设计 10](#_Toc137306341)

[2.5 本章小结 11](#_Toc137306342)

[3 仓库管理系统分析 12](#_Toc137306343)

[3.1 用户需求分析 12](#_Toc137306344)

[3.2 页面设计分析 13](#_Toc137306345)

[3.3 系统功能分析 14](#_Toc137306346)

[3.4 数据库分析 15](#_Toc137306347)

[3.5 可行性分析 18](#_Toc137306348)

[3.6 本章小结 19](#_Toc137306349)

[4 仓库管理系统实现 20](#_Toc137306350)

[4.1 开发环境 20](#_Toc137306351)

[4.2 主要功能模块实现 20](#_Toc137306352)

[4.2.1 后端项目的创建及配置 20](#_Toc137306353)

[4.2.2 页面搭建 22](#_Toc137306354)

[4.2.3 导航伸缩 23](#_Toc137306355)

[4.2.4 数据渲染 24](#_Toc137306356)

[4.2.5 分页展示 25](#_Toc137306357)

[4.2.6 增删改查 26](#_Toc137306358)

[4.2.7 登录验证 28](#_Toc137306359)

[4.2.8 路由管理 28](#_Toc137306360)

[4.2.9 权限管理 29](#_Toc137306361)

[4.3 主要功能测试 29](#_Toc137306362)

[4.4 本章小结 30](#_Toc137306363)

[结 论 32](#_Toc137306364)

[致 谢 33](#_Toc137306365)

[参考文献 34](#_Toc137306366)

[附 录 35](#_Toc137306367)

# 1 绪论

近些年来，伴随着互联网技术的发展以及信息化程度的提高，企业对仓库管理系统提出了更高的要求。仓库管理系统作为企业不可缺少的组成部分，它的功能不仅在于给企业提供了很好的管理平台，更在于对企业工作效率的提高，对成本的降低和市场竞争力的增强。

1.1 研究背景与意义

在企业规模越来越大的情况下，传统仓库管理方式已经不能满足企业越来越高的要求，软件产品需求量逐渐增加[1]。所以仓库管理系统也在逐渐朝着自动化、信息化的方向进行转变，从而顺应时代发展潮流。这一转变在促进仓库管理效率提高的同时，还能节省大量企业的人力、物力，从而降低管理的成本。

然而，当前市场上的仓库管理系统往往存在一些问题，例如前端页面交互体验差、后端代码复杂难维护等，这些问题导致了企业在实际使用过程中存在一定的困难。

基于此，本论文旨在研究一种基于SpringBoot和Vue的仓库管理系统前后端分离设计与实现的方案，SpringBoot是对Spring的再一次封装[2]，通过对前端、后端的分离，前后端分离的架构模式可使前端实现数据的交互、呈现，后端则负责业务逻辑的处理和数据的存取。这种架构模式可以提高系统的可拓展性、可维护性，也可以有效降低前后端代码耦合度，同时也有助于优化用户体验。SpringBoot框架作为整个系统架构模式，结合Mybatis-Plus框架实现持久层，采用分层思想实现[3]。

本研究对企业信息化转型，提升管理效率有重要的现实意义。一方面，可以帮助企业构建高效、稳定、易维护的仓库管理系统，提高管理效率，为企业带来经济效益。另一方面，本论文的研究也可作为相关领域学术研究的参考与借鉴，促进相关领域技术的发展与进步。

仓库管理系统作为一种重要的企业信息化应用系统，是管理企业仓库日常业务、实现物流管理的重要工具。在信息化技术日益发展的今天，仓库管理系统得到了越来越多的运用，对于提高企业管理效率，降低物流成本和增强客户满意度等方面都有着十分重要的意义。

然而，在实际开发过程中，传统的Web应用程序开发模式往往存在着前后端交互不够灵活、模块之间关联度（耦合度）过高等问题，导致系统的可维护性和可拓展性较差。为了解决上述这些问题，前后端分离的架构模式在近几年被广泛应用。鉴于此，本文的研究目的便是基于SpringBoot及Vue对仓库管理系统进行前后端分离设计与实现，希望能够为企业实现信息化转型有所帮助。

本文将对目前企业信息化转型趋势以及对仓库管理系统具体要求进行描述，阐述前后端分离开发的优势和该架构模式对于提高系统的可维护性和可拓展性的作用。通过详细介绍该系统的设计和实现方案，包括页面布局的拆分、组件化开发、增删改查功能的实现、权限管理实现等方面，使读者不仅能够了解前后端分离的架构模式，还能够掌握JavaWeb的开发技术。

此外，本文还将介绍基于B/S架构的Web应用程序开发模式，包括前端发起请求后，经SpringBoot的MVC架构（表现层、业务逻辑层和数据访问层）处理后，逐层返回结果给用户的流程原理。通过深入了解Web应用程序开发模式的深刻理解，使读者可以更好地理解系统的运行原理，以便于更好地进行开发和维护。

因此，本论文的研究对于促进企业信息化转型、提高企业管理效率、降低物流成本、提升客户满意度等方面具有重要的意义。

1.2 研究内容和贡献

本论文的研究内容是基于SpringBoot和Vue的仓库管理系统前后端分离设计与实现方案。具体而言，主要有以下几点：

1. 设计和实现一个基于前后端分离的仓库管理系统，包括管理员管理、用户管理、仓库管理和物流管理等功能模块的开发。

2. 基于ElementUI组件库，实现页面布局的拆分和组件化开发，提高用户体验和系统的可维护性。

3. 实现系统的增删改查功能和权限管理等功能，提高系统的可操作性和安全性。

4. 探究前后端分离架构模式的优势和作用，以及B/S架构模式在Web应用程序开发中的原理和流程。

本论文的主要贡献在于：

1. 研究了一种基于SpringBoot和Vue的仓库管理系统前后端分离设计方案，该方案充分利用了前后端分离架构的优势，将前端与后端解耦（降低不同模块间的关联性），提高了系统的可维护性和可拓展性。该方案在实现上采用了诸多技术，也实现了多个功能模块，且易于二次开发和升级。

2. 在仓库管理系统的页面布局设计和组件化开发方面进行了深入探究，将页面拆分成多个组件，实现了组件的复用，减少了代码冗余和维护成本。同时也通过采用ElementUI的丰富组件和样式库，提高了用户的体验。

3. 实现了增删改查和权限管理等功能，为企业信息化转型提供了一种可行的技术路线。

4. 对基于B/S架构的Web应用程序开发模式进行了详细介绍，为读者提供了一定的JavaWeb开发技术指导。

1.3 论文组织结构

本文共分为五个章节，具体内容组织如下：

第1章绪论：主要介绍本论文的研究背景和意义，以及研究内容和贡献。同时，还介绍了本文的组织结构和各章节内容的安排。

第2章关键技术介绍：介绍本文所用到的关键技术，包括后端开发技术、前端开发技术、B/S架构模式、前后端分离架构设计等方面。通过本章的介绍，读者可以了解到该系统所用到的关键技术以及技术背景。

第3章仓库管理系统分析：本章对仓库管理系统进行需求分析、页面设计分析、系统功能分析、数据库分析、可行性分析等，并对系统的实现进行概述。

第4章仓库管理系统实现：详细介绍系统的实现过程，包括开发环境的搭建、主要功能模块的实现、主要功能测试等方面。同时，还将介绍本文中提出的系统实现方案，以及系统的性能测试结果。

最后作为总结，将对本文所研究的基于SpringBoot和Vue的仓库管理系统前后端分离设计与实现方案进行了总结，并对系统的优缺点进行评价和分析。也评价和分析了该系统的优缺点，对未来研究的方向进行了展望、探讨。

# 2 关键技术介绍

本章将针对文中用到的相关技术做一综述，对基础的技术进行简述，着重介绍技术框架。前端技术概述包括了HTML、CSS、JavaScript、Vue.js和ElementUI等方面的知识点，这些技术都是Web前端开发必备的基础知识。后端技术概述主要介绍了Java语言及其相关技术，包括SpringBoot、Maven、MyBatis等。另外，在本章节中还会涉及到一些其他相关技术，例如前后端分离架构设计、B/S架构模式等。

2.1 后端开发技术

JavaWeb后端开发技术就是利用Java编程语言开发Web应用程序。JavaWeb技术的核心是Servlet和JSP技术。Servlet能够接收HTTP请求，该请求是来自浏览器的，并可响应相应HTTP响应。Servlet是个Java程序，运行在Web服务器中。JSP能够动态生成HTML页面的技术，它能将Java代码嵌入到HTML中。此外，JavaWeb后端开发还涉及到数据库的使用，开发人员需要掌握SQL语言和一种关系数据库管理系统（例如MySQL、Oracle等）。在JavaWeb应用程序中，开发人员通常使用JDBC（Java Database Connectivity）API访问数据库。JDBC API是一组被提供的接口与类，使各种数据库可以和开发人员进行交互，如处理结果集、SQL语句的执行和连接数据库。



图 2.1 Servlet架构

### 2.1.1 Java语言

Java是一款广泛应用于嵌入式系统开发、游戏开发、桌面应用程序、移动应用开发、企业级应用开发的跨平台的面向对象编程语言。最初由Sun公司开发并推广，它的设计目标是可移植性、简单性和高效性。Java语言的可移植性和对象性在Web编程中体现出了巨大的优势，两者结合产生了JavaWeb技术[4]。

### 2.1.2 SpringBoot

Spring框架是在J2EE的基础上实现的一个轻量级J2EE框架[5]。最初，企业级Java应用程序存在代码冗长、重复劳动和繁琐配置等各种问题，因此Spring应运而生，而如今它已成为一种开发生态圈，提供了若干子项目并各自完成特定的功能。Spring家族旗下的众多技术，其中最核心的便是SpringFrameWork，其他的技术都是基于SpringFrameWork的，它提供很多功能如依赖注入、事务管理、web开发支持等。Spring的依赖注入机制降低了业务对象替换的复杂性，提高了组件之间的解耦[6]。

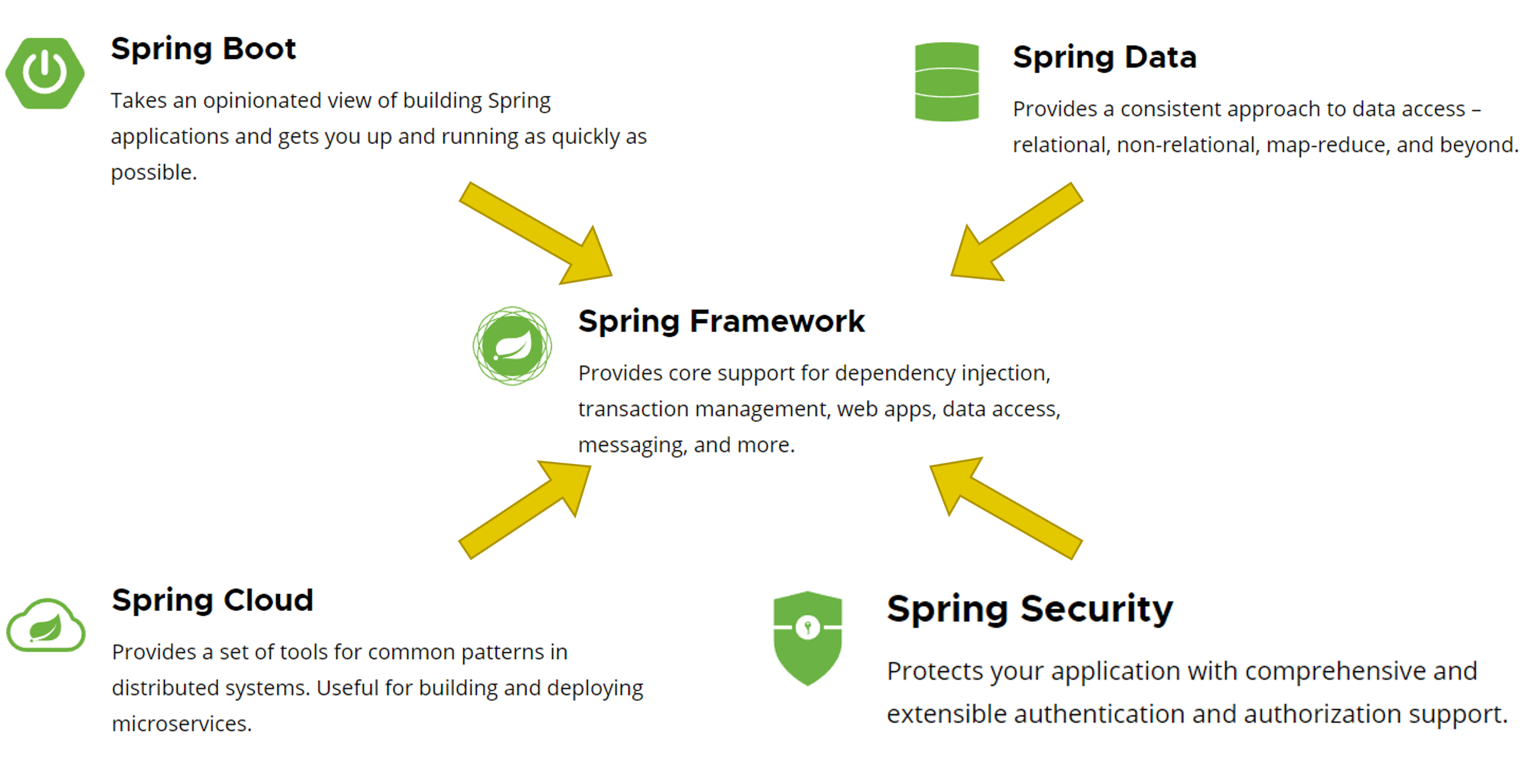


图 2.2 Spring技术关联

而若我们在项目中直接基于SpringFrameWork进行开发，配置繁琐的问题还是没有解决且入门难度较大。我们目前所使用的项目，是可以快速帮我们构建应用程序的项目SpringBoot。它提供的重要功能有很多，将Servlet容器内嵌于SpringBoot项目中便是其一功能。在Maven的配置文件pom.xml中引入spring-boot-starter-web的依赖之后，只要我们启动项目，便会自动启动Tomcat服务器。

在本校学习过凭借JavaEE构建一个简单的自定义web服务器处理客户端请求，会发现代码比较冗杂，大概流程是：

1.创建一个名为SS（ServerSocket）的对象，并调用这个对象的accept方法，之后等待来自客户端的连接。

2. 在线程的run方法中，通过输入流input获取客户端请求，输出流output向客户端发送响应。

3. 首先读取HTTP请求的第一行，若是GET请求，则将requestOK标记为true，继续读取HTTP请求的Header直到读取到空行。

4. 根据请求是否合法，发送相应的响应。若请求不合法，则发送404（Not Found）响应。如果请求合法，则读取指定的HTML文件，并将其转换为字符串，然后发送200（成功）响应，同时将HTML文件的内容作为响应体发送给客户端。

而在真正开发中用到web服务器时，都是使用目前比较流行的web服务器如Tomcat。web服务器是一个对HTTP协议的操作进行封装的应用程序，程序员就不必自己写代码去解析HTTP协议规则，让开发更加便捷。Tomcat是个开源免费的轻量级web应用服务器，支持Servlet/JSP规范。

前文讲到了，Servlet容器是内嵌在SpringBoot项目中的，因此在配置好Maven的web环境起步依赖后，启动项目成功时，Tomcat服务器也随即启动了。

代码清单 2.1 引入SpringBoot起步依赖

|  |
| --- |
| <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-web</artifactId> </dependency> |

在原始web程序中，若想在原始的web程序里获取请求信息，我们需要通过Servlet提供的HttpServletRequest（请求对象）API，见代码清单2.2。可以看到原始方式获取参数比较复杂，若在企业级开发中使用这种方式，代码会十分冗杂且不易维护。

代码清单 2.2 原始获取请求参数

|  |
| --- |
| public String getParam(HttpServletRequest request){  // http://localhost:8080/getParam?name=Jerry&age=18  // 请求参数: name=Jerry&age=18 （有2个请求参数）  // 第1个请求参数: name=Jerry 参数名:name, 参数值: Jerry  // 第2个请求参数: age=18 参数名:age, 参数值: 18  String name = request.getParameter("name");// 请求参数  String ageStr = request.getParameter("age");// 请求参数  int age = Integer.parseInt(ageStr);// 手动进行类型转换  System.out.println(name+" : "+age);   return "OK";  } |

以上方式不会在实际开发中使用，原因是SpringBoot环境对原始API进行了封装，使接受参数变得更加简单。若为简单参数，在形参变量名与简单参数相同的前提下，即可直接获取参数。若方法的形式参数与请求参数名称不相符，还想让方法中的形参想要接收到请求参数值，可以使用Spring提供的注解RequestParam注解完成映射。在方法形参前面加上RequestParam注解，再通过填充Value属性执行请求参数名而完成映射。

代码清单 2.3 SpringBoot方法获取请求参数

|  |
| --- |
| @RestController public class RequestController{  // http://localhost:8080/getParam?name=Jerry&age=18  // 请求参数名: name  @RequestMapping("/getParam")  public String getParam(@RequestParam("name") String username, Integer age){  System.out.println(username+" : "+age);  return "OK";  } } |

于控制层上方的RestController注解是组合注解，容纳了Controller和ResponseBody这两个注解。注解Controller类似注解Service和Repository，利用控制反转会创建一个代理对象Bean至IOC容器中，等待后续注入。ResponseBody注解则会使当前类下的所有方法的返回值作为相应数据，当实体对象或集合作为返回值时，注解会将其转换为JSON格式再给浏览器响应。类似JavaScript中的JSON.stringify方法，将对象实体转换为JSON字符串，再通过JSON.parse方法将JSON字符串转换为对象。

总的来说，SpringBoot是一个非常方便的框架，它能够帮助开发人员快速搭建Spring应用程序，并且具有高度的可维护性和可扩展性。关于SpringBoot的分层解耦，包括三层架构（SpringBootMVC）、控制反转（IOC）和依赖注入（DI）等，将会在后续章节“前后端分离架构设计”中详细阐述。

### 2.1.3 Maven

Maven是一款如今很流行的工具，用于管理、构建Java项目。它的功能主要有依赖管理、统一项目结构和项目构建等。依赖管理方面，Maven会对大型项目进行依赖管理，否则在开发项目的时候会有大量的jar包被引入。此时，jar包之间在很多时候需要版本匹配，否则容易出现版本冲突问题。当我们使用Maven进行jar包管理，便可以很方便地解决这个问题，只需在Maven项目中的pom.xml文件中进行配置即可。Maven工程不仅给我们自动生成了一个统一、标准的项目目录结构，避免项目目录结构不统一；也提供了自动化的项目构建方式，它通过一套简单的命令来完成项目的构建。

如图2.3所示，构建声明周期阶段，即完成标准化构建流程，在进行编译时，Maven为项目提供了编译插件；当我们需要打包时，Maven也提供了打包插件供使用。项目对象模型是我们自己的项目抽象出的对象模型，在pom.xml配置文件里我们可以看见自己项目专属的坐标，依赖管理模型描述了当前项目依赖的一些第三方jar包。

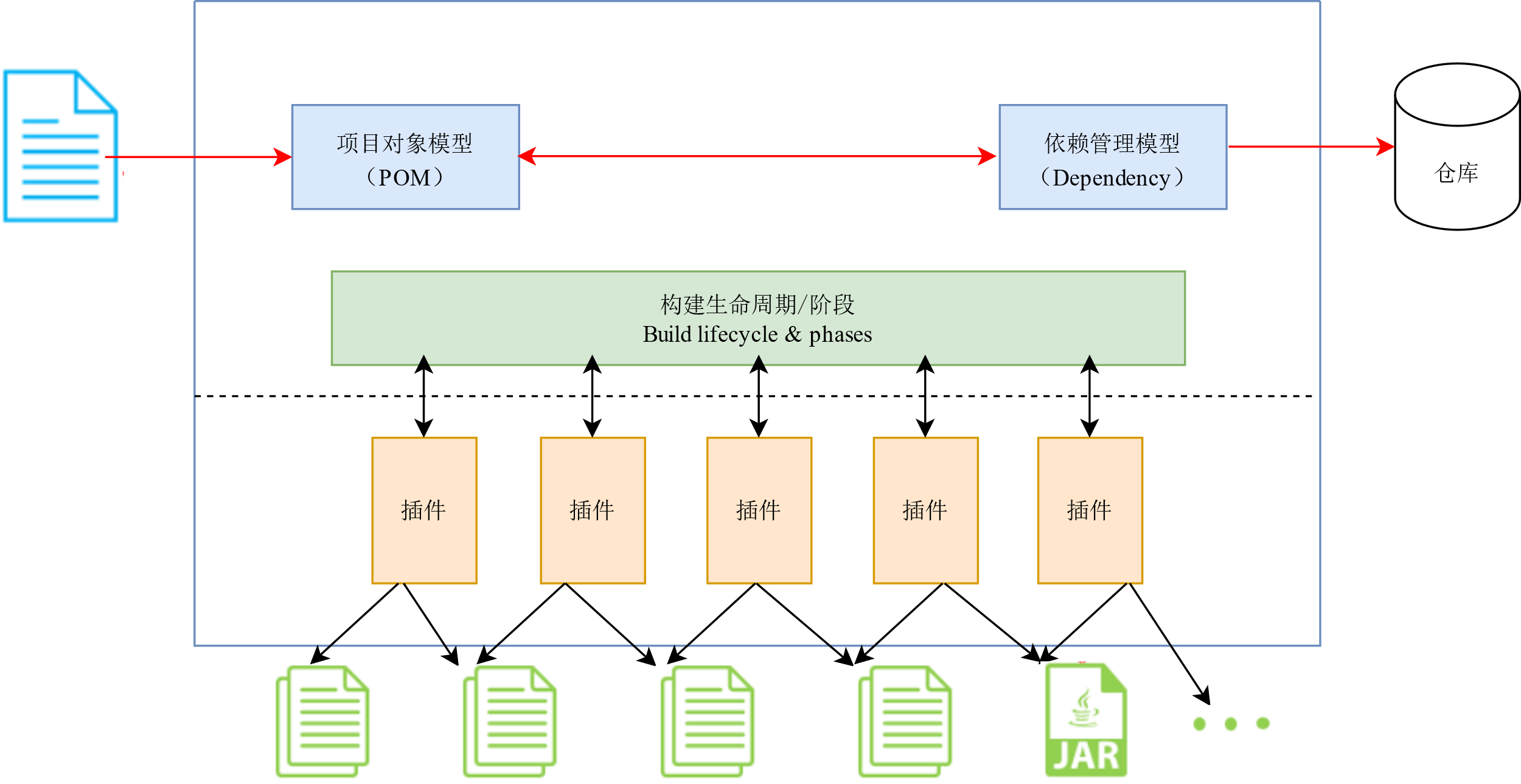


图 2.3 Maven模型

### 2.1.4 Mybatis

Mybatis是一款十分流行的用于简化JDBC（用Java语言操作关系数据库的一套API）开发的持久层框架，持久层即数据访问层。

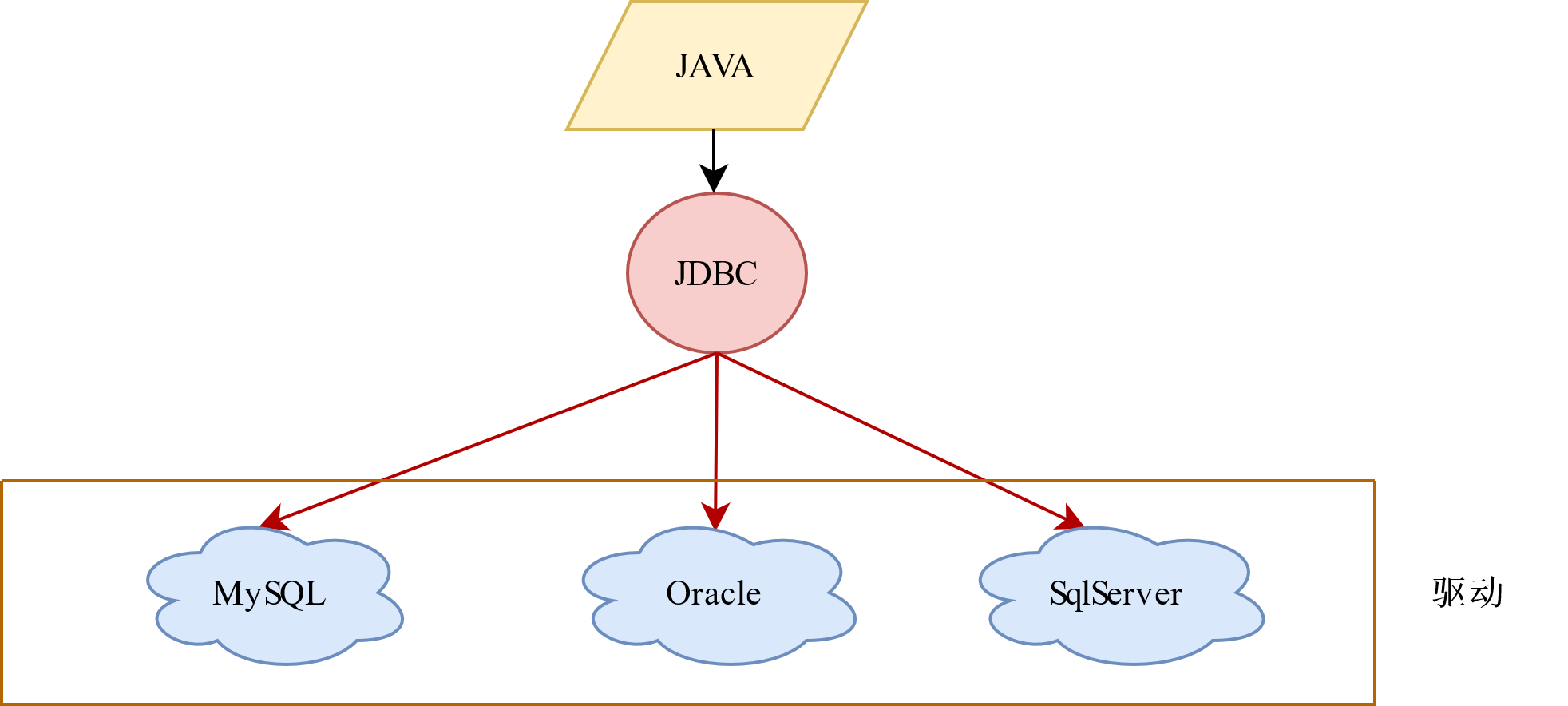


图 2.4 JDBC API

原始JDBC程序操作数据库的流程为：注册驱动、获取对象连接、释放资源、执行SQL语句、处理执行结果。这些操作用代码实现非常繁琐，因此在项目开发中，通常会使用Mybatis来操作数据库，从而简化数据库操作、提高开发效率、使Java应用程序能够轻松地与关系型数据库进行交互，提高了开发效率和代码质量。

2.2 前端开发技术

### 2.2.1 H5C3

HTML5、CSS3和JavaScript通常被合称为H5C3，是Web前端开发中常用的三种技术。HTML5作为最新HTML标准加入了许多新元素及API，使网页能够更丰富、更具有交互性。CSS3是CSS的最新标准，它加入了新的属性及选择器来更好地控制页面风格及布局。JavaScript是一种脚本语言，可以使网页具有更强的动态效果和交互性。

利用H5C3技术，可建立响应式、交互性好的网站及Web应用程序。HTML5具有较好的语义化结构以及多媒体支持等特点，能够实现网页内容在各种设备和屏幕中的自适应展示。CSS3则提供了更多的样式和布局选项，如动画效果、阴影和弹性布局等，可以让网页呈现出更加美观和专业的外观。而JavaScript则可以通过事件驱动、AJAX等技术实现网页的动态效果和实时交互。

H5C3技术被广泛应用于响应式Web设计，单页应用程序和游戏开发等现代Web开发。利用这些技术，开发者们能够创造出更好的用户体验，并提高网站的品质与价值。

### 2.2.2 Vue.js

Vue.js是一个轻量级的JavaScript前端框架，主要用于构建交互式的单页面应用程序。它采用了MVVM（Model-View-Model View）模式，利用数据绑定实现了前端和后端的数据同步更新。具有响应式的数据绑定、组件化开发、虚拟DOM、指令等特点，简化了开发人员的代码编写。其基于组件化的开发模式，使得开发人员可以快速构建复杂的应用程序，并实现了组件的可重用性，提高了开发效率。



图 2.5 MVVC框架

Vue的生态系统非常丰富，拥有许多插件和库，能够满足各种需求，例如vue-router实现前端路由、vuex实现状态管理、axios实现http请求等。综上所述，Vue是一个高效、易用、灵活可扩展的前端框架，具有良好的生态系统和活跃的社区，是构建现代单页面应用程序的一个不错的选择。

2.3 B/S架构模式

B/S架构是指Browser/Server架构，即浏览器/服务器架构模式。它是一种常见的分层模式，其中客户端（浏览器）作为一个客户端，通过互联网与服务器进行通信，并向用户提供服务。在B/S架构中，服务器负责处理业务逻辑，存储数据，而客户端负责显示用户界面和处理用户的请求。近年来，随着网络技术的迅速发展和J2EE平台的广泛采用，基于B/S的多层Web体系结构逐渐发展成熟起来，多层Web应用的开发已成为主流[7]。

B/S架构相比C/S架构，拥有许多优点，如使客户端不用安装额外的软件，这样就减少了维护成本和部署成本；能够实现跨平台、跨浏览器的兼容性；数据和业务逻辑集中在服务器端，提高了数据安全性和业务逻辑的可维护性和可扩展性等。因此，B/S架构在互联网应用开发中被广泛应用。

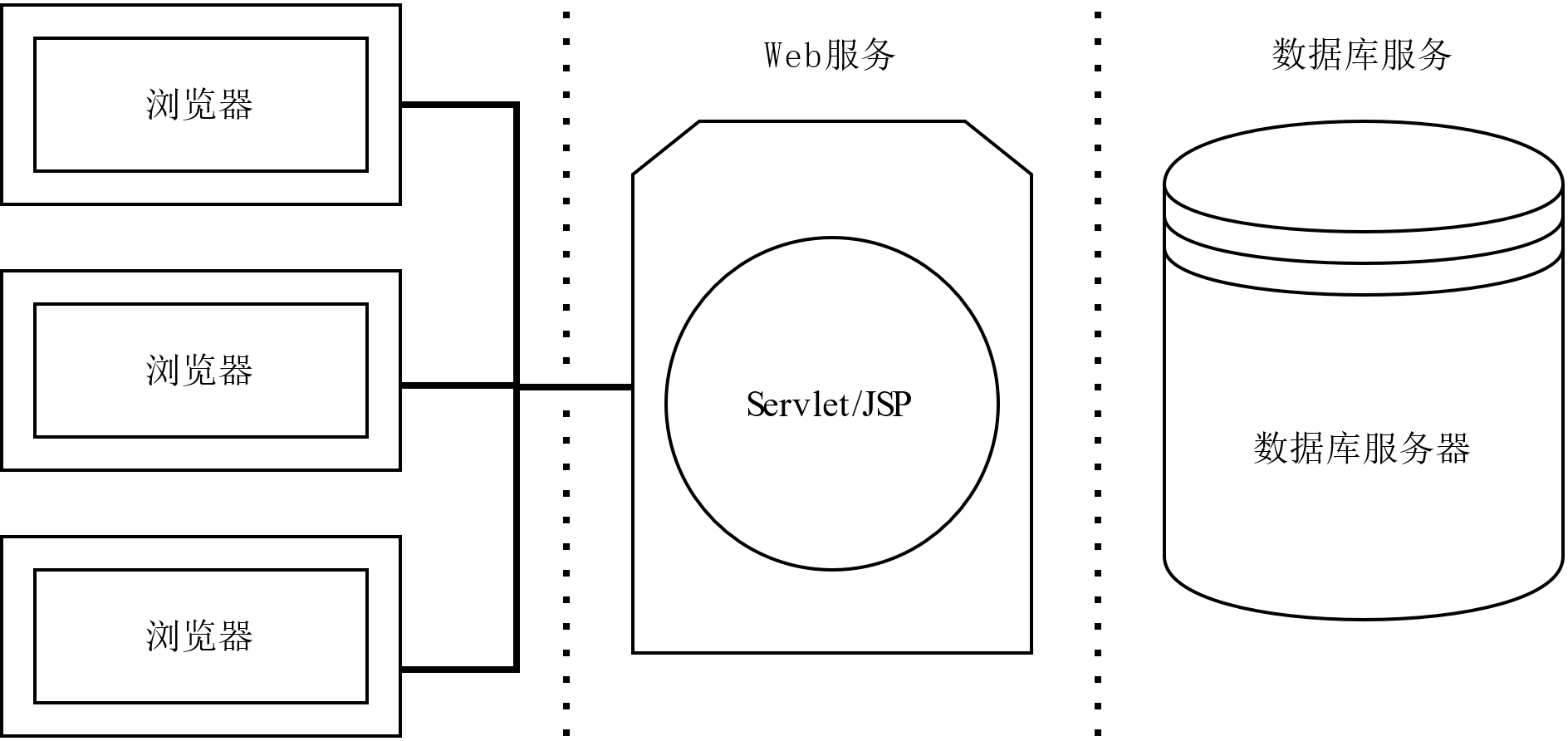


图 2.6 B/S架构

2.4 前后端分离架构设计

前后端分离架构是一种流行的Web应用程序设计方式，它将前端用户界面和后端业务逻辑的开发、测试和部署分开。这种架构可以提高开发效率、代码质量和可维护性，同时支持多种客户端（如Web、移动端、桌面应用等）的接入。

在前后端分离架构中，前端通常使用Vue.js、React等现代化的JavaScript框架来构建用户界面，后端则提供API接口，返回JSON或XML等数据格式，不再负责渲染页面。前后端之间通过HTTP协议通信，前端使用Ajax等技术来请求和处理数据，实现用户交互和数据展示。

在实现前后端分离架构的过程中，通常会采用SpringMVC三层架构模式。这种架构将Web应用程序分为控制层（Controller）、业务逻辑层（Service）和数据访问层（Data Access Object）三个部分。其中控制层负责接受前端发送的请求，对请求进行处理并响应数据；业务逻辑层负责处理业务逻辑和数据操作；数据访问层（Dao）也称为持久层，负责数据库的访问操作，包括数据的增、删、改、查。

除此之外，还需要使用控制反转（IOC）和依赖注入（DI）等设计模式，实现组件的解耦，方便维护和测试。IOC是指对象的创建控制权由程序自身转移到外部（容器），将组件之间的依赖关系交由框架管理，而不是自己手动创建和维护，从而减少代码的耦合性和冗余度；DI即提供应用程序运行时所需要依赖的资源，自动将组件所需要的依赖注入到其构造函数或属性中，而不需要手动创建和传递依赖。这种设计模式可以提高代码的可读性和可测试性，同时也方便了系统的扩展和维护。

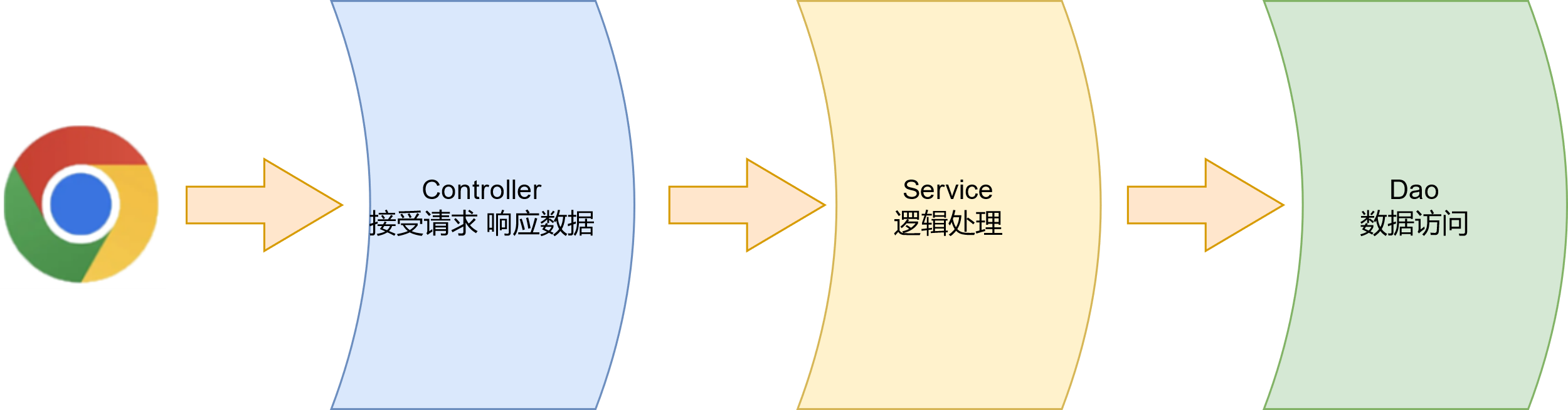


图 2.7 三层架构

2.5 本章小结

本章主要介绍了后端开发技术及前端开发技术两部分。在后端开发方面，我们选择了Java语言作为主要开发语言，并使用了SpringBoot框架进行快速搭建，以及Maven工具进行项目管理。Mybatis作为持久层框架用来操作数据库，实现对数据的增删改查等；在前端开发技术方面，我们使用了H5C3作为主要技术，用于实现网页的布局和美化，以及交互效果的实现。而Vue.js则是作为前端主要框架，用于实现组件化开发、数据的双向绑定和路由管理等功能。我们还简单介绍了B/S架构模式，以及前后端分离架构设计的相关内容，包括SpringMVC三层架构、IOC控制反转和DI依赖注入等技术。

通过本章的介绍，读者可以初步了解本项目所使用的技术栈和架构设计，有助于理解后文对管理系统设计和实现。

# 3 仓库管理系统分析

时下，仓库管理系统向数字化、智能化的方向发展，研究较为火热。伴随着物流与供应链行业的发展，仓库管理系统已成为企业物流管理与控制成本最主要的方法之一。当前，在数字化转型过程中，越来越多企业选择利用信息化技术对仓库管理进行提升与优化，从而提升效率并降低成本。

因此，针对当前企业的需求和行业的发展趋势，一个优秀的仓库管理系统应该具备数字化和智能化的特点，能够满足企业的管理需求，并支持各种新技术的集成和扩展。本章将从用户需求、页面设计、系统功能、数据库和可行性方面，对系统进行较为全面的分析。

3.1 用户需求分析

在用户需求这方面，企业预期的仓库管理系统是要能够满足其运营期望的，这个系统要有较为全面的管理功能，更要有足够的可维护性和可拓展性。具体来说，可维护性是指系统能够在用户访问量较高时保持稳定，避免系统宕机时出现崩溃导致数据丢失。可拓展性是指后端人员可以根据企业新增需求功能的描述，通过框架较为容易地进行功能的修改、删除和开发。从用户的角度，本文涉及的系统根据用户的不同身份，对功能进行了拆分，即普通用户和管理员可见功能数目是不一致的。为了满足用户对于系统的使用需求，我们对此系统有如下要求：

1. 界面简洁易用：系统操作界面要直观简单，尽量少用输入框及按钮，使用户可以很快上手。首页不仅要显示欢迎词，也要显示基本的个人信息及当前时间。

2. 功能齐全：此仓库管理系统，包括管理员和普通用户两种角色。管理员应该有权限对用户信息、仓库、物品分类、物品信息和出入库记录进行增删改查操作。普通用户只能对物品信息和出入库记录进行查询。

3. 数据安全性：本系统需保证系统的用户数据安全，登录界面要提供密码输入表单保证用户登录信息不泄露，表单提交使用的POST方法会使用户的登录数据会保存在请求体中，更加安全。

4. 系统稳定性：系统应具有高稳定性，能在用户访问次数较多的情况下稳定运行，以免系统宕机或者发生差错而造成用户数据的损失。这需要使用本系统的企业提供相应的服务器支持。

手机屏幕的截图

描述已自动生成

图 3.1 系统设计图

3.2 页面设计分析

仓库管理系统的前端页面设计要符合企业设计规范，即简洁且功能醒目易上手。本系统采用了基于Flex布局的ElementUI布局容器Container，方便快速搭建页面的基本结构。分为顶栏容器Header、侧边栏容器Aside和主要区域容器Main，在Vue中的文本结构类似HTML语言中的标签格式。Flex布局其采用“弹性盒子”模式，可实现不同屏幕尺寸及设备下自适应布置，使页面响应式设计更易于实施。

表格, 日历

中度可信度描述已自动生成

图 3.2 系统布局设计

在外层容器Container中，顶栏容器Header应能容纳欢迎词、个人用户名和左侧导航栏伸缩按钮（关联点击事件），且鼠标悬浮至个人用户名时，应能够显示一个下拉栏，展示“个人中心”和“退出登录”的功能。侧边栏容器Aside则应容纳首页、管理员管理、用户管理等页面的页面跳转按钮，后台通过Node.js已编写过路由跳转的功能，点击左侧对应模块即可跳转到对应页面。主要区域容器Main应显示对应数据表和分页模块，提供增删改查的表单和按钮供用户使用。

3.3 系统功能分析

系统功能分析就是通过对仓库管理系统的分析与提炼，从而确定出该系统所需的功能模块以及对应的操作流程。以下是针对该系统的系统功能分析：

1. 用户及权限管理功能

系统需要提供管理员管理功能，包括管理员的添加、编辑、删除和查询。其中，查询功能要求能根据姓名进行模糊查询，且能够根据性别分类查询。查询条件可以通过重置按钮一键清除。管理员需要控制登录权限，能够对系统进行用户管理、物品管理等操作。用户需要有登录权限，能够登录系统进行查询操作。

2. 仓库、物品分类管理功能

系统需要提供仓库管理、物品分类管理功能，包括仓库和物品分类的增删改查，且要求在点击添加后跳出的表单中有备注的文本框，用于标注该仓库或者物品分类的意义。

3. 物品管理功能

系统需要提供物品管理功能，包括但不限于增删改查的实现，在添加新物品时，要求在弹出的表单中，所属仓库和所属物品类别属性等信息要以下拉菜单的形式展示给管理员，在点击时会把数据库中现有的仓库和物品类别渲染到前端页面，以供选择。查询时要求能够根据不同仓库、不同物品类型或不同名称对物品进行查询。关于更改物品，该模块会提供出库与入库的按钮。通过选中某行数据（即某个物品），点击出入库按钮便会跳出表单，此表单中的申请人在点击后还会展示出子页面，用户可以根据子页面提供的查询功能，选中自己作为出入库的申请人。

4. 出入库记录管理功能

系统需要提供出入库记录查询功能，但要进行权限管理：管理员能够看到所有操作人员的操作记录，但普通用户只能查到和自己有关的出入库记录。此外，要求出入库查询模块展示的表格有按照时间排序的功能。

在本文中，我们已经详细列出了所需的系统功能模块。这些模块的实现需要依赖数据库的支持，因此我们需要进行数据库分析。数据库分析的目的是确定数据库的结构和组织方式。这涉及到表和关系的定义、数据类型的选择、索引与键的定义等方面。通过进行数据库分析，我们可以确保系统能够高效、准确地处理数据，并满足用户的需求。因此，数据库分析是实现系统功能的重要一步，这样的分析过程能够确保系统可以高效、准确地处理数据，并满足用户的需求。

3.4 数据库分析

数据库分析是指在系统设计之前对数据库进行分析和设计的过程，是系统设计的重要环节之一。数据库逻辑结构设计的质量将直接影响系统的运行效率及功能实现[8]。其必要性如下：

1. 明确数据结构与关系：数据库分析能够帮助开发人员更深入地理解系统内数据结构与数据间的关系，从而使得系统内数据更清晰、有序、可维护、可扩展。

2. 保证数据完整性与一致性：通过数据库的分析与设计，可清楚地了解各表间关系，数据类型及数据格式，保证系统内数据的输入、存储、输出等环节具有一致性和完整性。

3. 提高数据处理效率：对数据库进行优化，例如建立索引、使用合适的数据类型等，可以大大提高数据处理效率，减少系统响应时间。本项目的数据库数据量不大，并未索引。但是在企业级的业务中，数据可能达到千万的量级，这时候使用查询语句会花费数十秒的时间，效率及其低下。通过使用索引优化，可以将查询时间优化到毫秒级，这是优化效果是极其显著的。

4. 降低系统开发和维护成本：通过数据库分析，可以减少因数据结构和关系不清晰而导致的开发和维护成本，使系统更加可维护、可扩展。

5. 为系统安全提供保障：通过数据库分析，可以制定合适的数据访问策略，限制不同用户的数据访问权限，从而提高系统的安全性。

因此，数据库分析对于项目的开发和维护都具有重要的必要性，是开发一个高效、可靠、安全、易维护的系统的必要步骤，良好的数据库分析可为功能模块的开发提供方向。接下来，本文根据涉及系统的六张表：用户表、菜单表、仓库表、物品分类表、物品表、记录表进行分析。

1. 用户表

该表用于存储系统用户的信息，包括id、账号、名字、密码、年龄、性别、电话、角色以及是否有效等字段。该表的主键是id字段，并设定为自增。因为id与用户是一一对应的，即使某个用户被删除，新增用户也会从目前最靠后的id开始自动生成。其中，角色（roleId）字段为角色表的外键，用于标识用户的角色，通过roleId的不同int值来划分用户的权限：0代表超级管理员，1代表管理员，2则代表普通用户。是否有效（isValid）字段为表示该用户账号目前是否有效，“Y”代表有效，其他varchar值为无效，并设定默认值为“Y”，即有效。

表 3.1 用户表设计

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 非空 | 主键 | 默认 | 备注 |
| id | int | 11 | true | true | auto\_increment | 主键 |
| no | varchar | 20 |  |  | null | 账号 |
| name | varchar | 100 | true |  |  | 名字 |
| password | varchar | 20 | true |  |  | 密码 |
| age | int | 11 |  |  | null | 年龄 |
| sex | int | 11 |  |  | null | 性别 |
| phone | varchar | 20 |  |  | null | 电话 |
| role\_id | int | 11 |  |  | null | 角色 |
| isValid | varchar | 4 |  |  | Y | 是否有效 |

2. 菜单表

该表用于存储系统菜单的信息，包括菜单编码、菜单名称、菜单级别、菜单父编号、点击触发的方法名、权限、组件路径和图标路径等字段。该表主键为id字段，自动增长。其中，菜单父编号字段用于标识菜单之间的层级关系，本系统的菜单栏设置为同级，所以暂用不到该字段。若企业希望左侧菜单栏能多层展示，可以在Vue项目中修改，并填充该字段内容即可。权限字段用于标识菜单对应的访问权限，“0”、“1”、“2”分别对应超级管理员、管理员和普通用户，可以逗号分隔组合填充。组件和图标字段用于页面的展示和操作。

表 3.2 菜单表设计

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 非空 | 主键 | 默认 | 备注 |
| id | int | 11 | true | true | auto\_increment | 主键 |
| menuCode | varchar | 8 |  |  | null | 菜单编码 |
| menuName | varchar | 16 |  |  | null | 菜单名 |
| menuLevel | varchar | 2 |  |  | null | 菜单级别 |
| menuParentCode | varchar | 8 |  |  | null | 父菜单 |
| menuClick | varchar | 16 |  |  | null | 绑定事件 |
| menuRight | varchar | 8 |  |  | null | 权限 |
| menuComponent | varchar | 200 |  |  | null | 模块路径 |
| menuIcon | varchar | 100 |  |  | null | 图标 |

3. 仓库表、物品分类表

该表用于存储系统中的仓库信息，包括仓库编号、仓库名称和备注等字段。主键是自动增长的id字段。仓库名称字段用于标识不同的仓库，备注字段用于存储仓库的相关说明。物品分类表设计同仓库表，本文不过多赘述。

表 3.3 仓库表设计

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 非空 | 主键 | 默认 | 备注 |
| id | int | 11 | true | true | auto\_increment | 主键 |
| name | varchar | 100 | true |  |  | 仓库名 |
| remark | varchar | 1000 |  |  | null | 备注 |

表 3.4 物品分类表设计

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 非空 | 主键 | 默认 | 备注 |
| id | int | 11 | true | true | auto\_increment | 主键 |
| name | varchar | 100 | true |  |  | 分类名 |
| remark | varchar | 1000 |  |  | null | 备注 |

4. 物品表

该表用于存储货物的详细信息，包括货物编号（自增主键）、货物名、所属仓库id、所属分类id、现货数目、备注等字段。其中，该表与仓库表和物品分类表存在关联，仓库表和物品分类表的id字段，是该物品表的逻辑外键。

表 3.5 物品表设计

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 非空 | 主键 | 默认 | 备注 |
| id | int | 11 | true | true | auto\_increment | 主键 |
| name | varchar | 100 | true |  |  | 物品名 |
| storage | varchar | 11 | true |  |  | 仓库名 |
| goodsType | varchar | 11 | true |  |  | 分类名 |
| count | int | 11 |  |  | null | 数量 |
| remark | varchar | 1000 |  |  | null | 备注 |

5. 记录表

该表用于存储系统中的操作记录，包括操作记录编号、货品编号、取货人/补货人编号、操作人编号、出入库货物数量、操作时间和备注等字段。该表主键为id字段（自增）。其中，货品编号字段用于标识操作记录对应的货品，取货人/补货人编号字段用于标识申请出入库的用户，操作人编号字段用于标识执行批准操作的管理员，数量字段标识货品的数量变化，操作时间字段用于记录操作的时间，备注字段用于存储操作记录的相关说明。其中，物品表和用户表中的id字段，也是该记录表的逻辑外键。

表 3.6 记录表设计

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 非空 | 主键 | 默认 | 备注 |
| id | int | 11 | true | true | auto\_increment | 主键 |
| goods | int | 11 |  |  | null | 货物id |
| userId | int | 11 |  |  | null | 取/补货人 |
| admin\_id | int | 11 |  |  | null | 操作人id |
| count | int | 11 |  |  | null | 数量 |
| createTime | datetime |  |  |  | current\_timestamp | 操作时间 |
| remark | varchar | 1000 |  |  | null | 备注 |

在分析上述表的同时，有提到逻辑外键和物理外键，逻辑外键和物理外键是两种不同的关系约束方式。

逻辑外键在表结构设计中通过字段值之间相互联系达到对关系进行约束的目的，不需要借助数据库引擎对其进行约束。逻辑外键的实现依赖于应用程序对数据的读取和写入进行控制，通常需要在应用程序中手动实现关系的检查和维护。逻辑外键具有无需对数据库结构进行修改等优点，但是其缺点是应用程序中要写很多逻辑代码以保持关系完整，同时也容易出错。

物理外键通过数据库引擎实现了对关系的限制，外键关系维护交由数据库引擎进行。在创建表时，使用外键约束将两个表关联起来，当一张表的主键被更新或删除时，对应的外键也会被自动更新或删除，从而保证了数据的完全性、一致性。物理外键的优点在于可以通过数据库引擎自动维护数据完整性，而且也比较容易理解和使用，缺点则是增加了数据库的复杂度，有时会对性能产生一定的影响。

因此，逻辑外键和物理外键的选择取决于具体的应用场景和需求。如果数据模型比较复杂，或者需要保证数据的一致性、完全性，那么物理外键是种合适的选择；若数据模型相对简单，或需经常执行更新，插入和删除等操作而对数据完整性又无特别强烈的要求时，逻辑外键不失为一种较好方法，本系统关联的数据表采用的便是逻辑外键。

3.5 可行性分析

可行性分析在项目前期起着举足轻重的作用，它主要从技术，经济和市场三个层面来评价项目是否可行。在本文中，进行以下方面的可行性分析：

1. 技术可行性

技术可行性是对待开发系统进行功能、性能和限制条件进行分析，对在当前的技术条件下系统能否实现，实现开发系统的风险有多大[9]。该仓库管理系统采用了目前比较成熟的技术架构，前端使用了Vue和ElementUI组件库，后端使用SpringBoot框架，数据库采用MySQL。这类技术均有比较广泛的应用与支撑，具有开发难度小、开发效率高等特点。该系统所需硬件配置低，普通台式电脑或笔记本电脑均可达到使用需求，故硬件可行性亦较强。

2. 经济可行性

开发本系统所需的开发工具和软件基本上都可以免费获取或者采用开源软件，因此可以在低廉的开发成本下完成开发。并且，本系统能对仓库存储和出入库记录进行有效管理，避免了因存货过多而造成的滞销、浪费等问题，可以在一定程度上提高仓库管理的效率和企业利润，因此在经济可行性方面也具有潜在的优势。

3. 市场可行性

仓库管理作为企业日常经营中最重要的一个环节，所以该项目的市场需求较大。同时，当前越来越多的企业开始采用互联网技术来优化和提升仓库管理效率，这也为该项目的市场推广提供了较好的契机。因此，从市场角度来看，该项目也具有可行性。

综上所述，从技术、经济、市场等方面来看，该仓库管理系统具有可行性，可以继续开展后续的开发和实施工作。

3.6 本章小结

这一章主要是对仓库管理系统进行分析和设计。首先，我们进行了用户需求分析，了解了用户的需求，从而确定了系统的功能模块。然后，我们进行了页面设计分析，根据用户需求设计了用户界面，使得用户操作系统更加简单、直观。接着，我们进行了系统功能分析，将系统的功能模块进行了细化，并对模块之间的交互关系进行了说明。接下来，我们进行了数据库分析，对系统中涉及到的数据进行了设计，确定了数据表和字段。最后，我们进行了可行性分析，对系统的实现可行性进行了评估，确定了系统的实施方案。通过本章的分析与设计，我们明确了系统的功能模块、界面设计、数据库设计等内容，为系统的实现提供了有力的支持和保障。

# 4 仓库管理系统实现

4.1 开发环境

通过上一章的分析与设计，我们了解了系统的功能需求和设计思想。本章会具体说明系统的功能实现过程，以及进行相应模块功能的测试。首先，要充分准备好开发环境，以便后续的开发和调试。

表4.1 开发环境

|  |  |
| --- | --- |
| 开发工具 | IntelliJ IDEA 2022.2.4，Visual Studio Code 1.77.3 |
| 接口测试工具 | Postman Agent |
| 数据库 | MySQL 5.6.43 |
| 数据库图形化 | Navicat Premium 15 |
| 前端调试工具 | Google Chrome 112.0.5615.138 |
| 开发包 | SpringBoot，Vue.js、Maven，Mybatis，ElementUI，Lombok |

4.2 主要功能模块实现

本节会描述系统的主要功能模块的实现过程，包括前端和后端的实现细节。详细介绍一些重要功能模块的实现逻辑和代码结构，包括页面搭建、数据渲染、增删改查、分页展示、路由管理、登录验证、导航伸缩、权限管理等功能的实现过程。

### 4.2.1 后端项目的创建及配置

作为一个前后端分离的项目，首先要先利用开发工具Idea创建一个SpringBoot项目，并在该项目的Maven配置文件pom.xml中添加项目需要用到的依赖，如Mybatis-Plus、MySQL、Lombok等，并配置好阿里云镜像加速依赖下载并装载。接着，要根据第三章的数据库分析，创建数据库和表格并插入原始数据。还要在yml配置文件中配置数据库驱动类名称、MySQL账号密码和项目启动占用的端口号。在这之后，我们要在项目包目录下创建一些包，内含common、controller、entity、mapper、service等。其中，common是存放配置类的包，包括解决跨域问题、MyBatis-plus、查询分页的配置文件和返回数据的分装Result类。controller层（接受请求和返回响应）和service层（编写业务逻辑）以及mapper层（操作数据库）的三层架构在第二章已做说明，此处不再赘述。实体类包内含一些系统需要用到的与数据库表相关的数据模型，在实体类中的“@Data”注解是由Lombok依赖提供的，会自动生成对应属性的GET和SET方法以及构造函数。除此之外，还要在资源包目录下创建同名的mapper包并在内部放置mybatis的xml配置文件，这些配置文件对应着项目包com.back下的mapper类，可以将SQL的查询结果直接返回给相应方法。

图形用户界面

低可信度描述已自动生成

图 4.1 后端项目结构

由于我们加入了Mybatis-Plus依赖，在创建mapper接口时，要继承BaseMapper<T>（T代表泛型，后文出现不再说明），并且在对应接口的Mybatis配置文件里编写对数据库的操作。创建service接口时要继承IService<T>，创建service实现类时，也要继承ServiceImpl<T,T>。在实现类中注入mapper后，若业务具有需求，我们可以自行添加自定义方法。但Mybatis-Plus内置了诸如增删改查等方法，我们也可以直接调用。

以用户管理为例，我们首先要根据数据库的用户表字段，在实体类中创建User用户实体类User并通过Data注解生成相应属性的GET和SET等方法，接着便要创建Mapper接口userMapper、创建Service接口userService和对应实现类UserServiceImpl。然后在controller包下创建userController类，注入相关Service类，声明系统需求的方法如增删改查等。方法可以调用Mybatis-Plus自带的方法，也可以使用自定义方法。自定义方法需要在mapper类中声明自定义方法名，并在Mybatis的xml配置文件中编写该方法对应的SQL语句。

以上便是后端接收前端请求响应，经过控制层、业务处理层和DAO层访问数据库，并将处理返回结果逐层提交返回给前端的流程。本项目返回给前端的对象是Result类，是对数据的封装，方便统一处理。其属性有code、msg、total和data，分别代表编码（200表示成功，400表示失败）、提示信息（成功或失败）、总记录数和数据（对象格式）。后端涉及的方法包括增加、删除、更改、分页查询、模糊查询、精确查询和用户登录等。这些代码的实现逻辑是一样的，本文不做重点描述，接下来会重点阐述基于关联这些后端方法Vue框架对该仓库管理系统功能开发。

### 4.2.2 页面搭建

本系统的前端页面是基于Vue框架和ElementUI组件库搭建的，页面结构被拆分为头部、侧边栏和页面主体三部分，详见3.2页面设计分析。要使用ElementUI组件库，我们需要先安装Vue脚手架Vue/Cli，它是专门为单页面应用实行快速搭建的。接着，我们还要下载并安装好Node.js，配置环境变量。接着，我们在项目文件夹下开启终端并输入“vue create 项目名”并选择Vue2.0来创建项目。

图片包含 文本

描述已自动生成

图 4.2 前端项目结构

在项目创建完毕后，便可以开始前端页面的开发了。首先我们在组件文件夹components下创建index.vue，在template的div标签中放入ElementUI的布局容器组件，并在style标签下设定样式，在script标签的export default（对外输出模块）的data对象中放入初始数据。接着在app.vue这个项目入口中导入Index并开启项目，可以看到ElementUI的布局容器已经可以正常显示了，但有时需要我们手动调试页面样式，可以在assets目录下新建global.css并在main.js中引入，使其作为全局样式文件并在项目启动时被自动加载。

代码清单 4.1 页面布局结构

|  |
| --- |
| <el-container>  <el-aside></el-aside>  <el-container>  <el-header></el-header>  <el-main></el-main>  </el-container>  </el-container> |

ElementUI提供的默认布局容器各模块之间的耦合性过高，不利于项目的维护和拓展，且代码过于冗杂，可读性不高。因此，我们要对页面进行拆分：在components文件夹下创建Header、Aside和Main的vue文件，将原本index.vue下各模块的代码转移到对应新建的三个vue文件中。并在index.vue下将这三个vue文件作为组件引入，填充至container中对应的标签内（如在el-aside标签下填充<Aside></Aside>），页面拆分就此完成了。头部区域Header使用Flex布局，分为左中右三部分。左侧是导航栏伸缩按钮，中间是系统名称，右侧是用户名称并引入了下拉菜单，有个人中心和退出登录功能。

### 4.2.3 导航伸缩

上文提到头部区域的左侧存在一个导航栏伸缩按钮，那么它如何实现左侧导航栏的伸缩功能呢？接下来会阐述实现思路：首先在Aside.vue中的el-menu（导航栏模块）中有一个默认的属性值collapse（默认值为false），我们要给Header中的伸缩按钮绑定一个点击事件，这个事件会提交请求方法名（如doCollapse方法）给父组件index，一旦按钮监测到被点击，父组件index便会执行doCollapse方法。我们定义这个方法的作用是通过Vue的数据绑定v-bind将Aside导航栏el-menu的collapse属性取非（即false变为true，true变为false），这样在每次点击伸缩按钮时，左侧导航栏则会收缩与展开了。

### 4.2.4 数据渲染

在获取后端数据渲染到前端页面的过程中，首先我们要安装axios并全局引入到main.js文件中。但在项目启动获取数据时，页面调试工具的NetWork模块中会报错“strict-origin-when-cross-origin”。这是由浏览器的同源策略（双方的协议、域名、端口都要相同，有一个不同都会产生跨域）导致的，初衷是为了防止用户的数据被恶意网站盗用而对JavaScript施加了安全限制，一般称其为跨域问题。

为了解决这种问题，本文采用重写WebMvcConfigurer（全局跨域）的方式。在后端项目包下的common包下创建CrosConfig类并编写跨域代码，即可解决跨域报错问题。

代码清单 4.2 解决跨域问题的配置

|  |
| --- |
| @Configuration public class CorsConfig implements WebMvcConfigurer {  @Override  public void addCorsMappings(CorsRegistry registry) {  registry.addMapping("/\*\*") //是否发送Cookie  .allowCredentials(true) //放⾏哪些原始域  .allowedOriginPatterns("\*")  .allowedMethods(new String[]{"GET", "POST", "PUT", "DELETE"})  .allowedHeaders("\*")  .exposedHeaders("\*");  } } |

现在可以正式将后台数据库数据渲染至前端页面，并以表格形式呈现。首先我们要在Main.vue中引入el-table标签，这是ElementUI组件库的表格组件，列标签是el-table-colomn。每个列标签都有一些常用的属性，例如prop（对应后台数据字段名）、label（前端页面展示的列名）和width（列宽）。并要在el-table中设置单向绑定的值“tableData”，这是一个用于保存对象的数组，后台将会获取对象数据存放到这个数组中。我们要在挂载函数中（页面加载时就启动的一些方法）添加GET和POST方法，这样在页面加载时数据会自动从后端获取并存储到数组tableData中，Vue框架会把这个数组中的对象数据以el-table表格形式展示出来，这便是数据渲染的大致流程。

其次，我们需要将后台获取的某些数据转换为另一种形式予以展示给用户。例如用户表中的sex字段的值为0或1，我们不可能在页面上以数字形式展示，这样并不美观且不符合业务要求。我们可以在性别所在的列标签中使用el-tag标签使用template插槽，其中slot-scope的作用是可以使用子组件中的数据，通过令slot-scope=“scope”来取得作用域插槽data绑定的数据。本系统设置sex值为0会以success样式的“女”显示性别，而1会以primary样式的“男”显示。

代码清单 4.3 Tag转换列

|  |
| --- |
| <template slot-scope="scope">  <el-tag  :type="scope.row.sex == '1' ? 'primary' : 'success'"  disable-transitions>{{scope.row.sex == '1' ? '男' : '女'}}  </el-tag>  </template> |

最后，我们在数据表的操作列预添加“编辑”和“删除”的按钮，等待功能的开发并在后期给按钮绑定事件即可。还可以给数据表添加样式，如背景颜色和边框等。这样初始数据表格的搭建和渲染就完成了。

### 4.2.5 分页展示

本节会使用到el-pagination分页组件，首先要在el-table标签的下方将其引入。分页组件也内含一些常用属性，包括page-sizes（每页显示个数选择器的选项设置）、current-page（当前页数）、layout（组件布局，子组件用逗号分隔）、page-size（当前页面最大容纳记录个数）、total（总记录数）和绑定的方法名等。

分页组件绑定了两个方法，分别是“handleSizeChange”和“handleCurrentChange”，方法逻辑很简单：根据用户点击组件选择的数据或页数，更新表格的单页条数和当前页属性，并刷新表格。由于在后端已经编写了分页查询的方法listPageC1，因此在main.vue中的POST方法中调用的也是分页查询的方法。分页组件绑定的两个方法的逻辑是相同的，下面仅展示其中一个以及POST方法，POST方法中通过使用axios的方法异步请求数据，其中的$httpUrl是在main.js中设置的全局后端地址。

代码清单 4.4 分页组件主要方法

|  |
| --- |
| handleSizeChange(val) {  this.pageNum = 1  this.pageSize = val  this.loadPost()  },  loadPost() {  this.$axios.post(this.$httpUrl + '/user/listPageC1', {  pageSize: this.pageSize,  pageNum: this.pageNum,  param: {name: this.name, sex: this.sex}  }).then(res => res.data).then(res => {  //若相应状态为200, 则获取数据和总记录数  if(res.code == 200){  this.tableData = res.data  this.total = res.total  }else {alert('获取数据失败!') }})  } |

### 4.2.6 增删改查

本节会基于对用户表介绍该管理系统的增删改查等操作的实现：

1. 传参查询功能

为了实现传参查询功能，我们在el-table上方新增了一个el-input标签，该标签作为输入框，并绑定了回车事件来调用POST方法。同时，在输入框的右侧添加了查询和重置按钮，并给查询按钮添加了点击触发事件来绑定POST方法。该方法是由后端编写的查询操作，并在前端根据查询结果来刷新页面。为了增加按男女查询的功能，我们还在输入框的右侧添加了一个下拉菜单el-select，其属性el-option中的v-model（双向绑定）值为“sex”。当用户选择男或女时，经过Vue双向绑定的sex值会流向后端进行精确查询，并返回给前端页面刷新查询结果。最后，我们给重置按钮绑定了一个方法，该方法可以清空表单和下拉菜单中的数据，以实现重置查询条件的功能。

2. 新增功能

在重置按钮右侧新增添加按钮，并为其绑定一个点击事件，该事件将调用一个名为add的方法。接下来，在el-table表格组件下方新增一个el-dialog对话框组件，并设置其属性centerDialogVisible的初始值为false。我们需要编写add方法来将这个属性设置为true，以实现点击添加按钮后显示对话框的效果。然后，在el-dialog组件中引入el-form表单组件，并将每个表单项与一个双向绑定的表单值关联起来。

代码清单 4.5 校验规则

|  |
| --- |
| rules: {  no: [  { required: true, message: '请输入账号', trigger: 'blur' },  { min: 3, max: 16, message: '长度在 3 到 16 个字符', trigger: 'blur' },  {validator:checkDuplicate,trigger: 'blur'}],  name: [  { required: true, message: '请输入姓名', trigger: 'blur' }],  password: [  { required: true, message: '请输入密码', trigger: 'blur' },  { min: 3, max: 16, message: '长度在 3 到 16 个字符', trigger: 'blur' }],  age: [  {required: true, message: '请输⼊年龄', trigger: 'blur'},  {min: 1, max: 3, message: '⻓度在 1 到 3 个位', trigger: 'blur'},  {pattern: /^([1-9][0-9]\*){1,3}$/,message: '年龄必须为正整数字',trigger: "blur"},  {validator:checkAge,trigger: 'blur'}] |

为了提交用户编辑的数据，我们需要在对话框底部添加确认和取消按钮。点击取消按钮将centerDialogVisible设置为false，关闭对话框并退出添加功能；点击确认按钮则会调用一个名为save的方法。该方法已经在后端项目中编写完成，使用$axios.post方法发送请求，与POST方法相同。如果添加操作返回的Result状态值（res.code）为200，则关闭对话框，并通过ElementUI提供的消息提示组件Message提示添加成功；否则，将弹出添加失败的消息提示。最后，我们需要为新增数据的表单添加格式校验，使用el-form组件中的单向绑定rules属性，可以在script标签中的data对象中编写校验规则。

3. 编辑功能

之前在表单的操作列中，我们添加了编辑和删除的按钮等待功能开发，我们只需在编辑按钮el-button中添加点击事件，绑定一个名为mod的方法（关联后端的update更新方法），同添加操作设置一个带有表单的对话框，提交表单数据到后台修改数据库对应数据项，并重新渲染页面。

4. 删除功能

删除功能的实现相对与查增改较为简单，只需在操作列对应的template添加插槽，给删除按钮添加绑定事件关联del方法（关联后端的del删除方法），根据id删除数据库对应的数据项，并重新渲染界面。

### 4.2.7 登录验证

实现登录验证需要新建Login.vue文件，该文件包含一个大的div盒子，其中包含h1标题、el-form表单（含验证规则）和el-button确定按钮，并编写了相应的页面CSS样式。当点击确定按钮后，会触发confirm方法进行登录验证。在此过程中，该方法会将确认按钮的confirm\_disabled属性设置为true，以避免表单重复提交。接下来，该方法会调用后端的login方法进行用户数据验证查询。如果查询到用户，则会返回状态码200，并通过sessionStorage.setItem方法将登录用户的数据存储到本地缓存中，并跳转到首页；否则，会弹出消息“校验失败，用户名或密码错误！”。

### 4.2.8 路由管理

要实现路由管理，我们需要安装vue-router插件并创建路由文件。在main.js中引入vue-router，并将app.vue中的<Index/>组件改为<router-view/>，这样就可以使用vue-router进行路由管理了。我们还需要为退出登录和个人中心功能设置路由。在个人中心中，我们可以从缓存中获取已登录用户的用户名，然后通过下拉菜单中的“退出登录”菜单项执行logout方法，这个方法将路由跳转回登录页面并清空本地缓存。对于“个人中心”菜单项，我们只需将路由跳转到UserHome组件即可。为了在Aside导航栏中使用v-for语句生成动态菜单，我们需要在menu对象中添加菜单对应组件名、菜单名和菜单图标名。这也为后文的权限管理做好了准备。

### 4.2.9 权限管理

权限管理是基于上一章数据库设计中的菜单表设计的，根据不同用户的权限来生成不同的菜单栏。我们首先要构造出表并插入数据如管理员管理、用户管理、仓库管理、物品分类管理、物品管理和记录管理的功能模块。并指定超级管理员可操控所有模块，普通管理员可以操控除管理员管理之外的所有模块，普通用户只可使用物品管理和记录管理模块。接下来，我们要在后端编写好菜单查询的代码，在登录之后会根据用户权限返回查询到的菜单模块信息，并在Aside侧边栏通过v-for语句动态生成菜单。为了解决多组件之间数据通信的问题，我们还要安装vuex并编写store文件并在main.js中引入。在Login.vue中通过语句“this.$store.commit”提交用户可操作的菜单数据，在任意组件中，可以通过this.$store.state来获取公共数据，因此只需在Aside.vue中的computed中通过get方法得到store公共数据的菜单数据后，生成相应菜单即可。

另一权限管理的需求首先是用户只能提交对物品出入库的申请，并不能手动增加物品信息和出入库。其次是管理员可以看到所有用户的出入库信息，但是普通用户只能看到自己的出入库信息。第一个需求只需在相应的功能按钮组件上通过v-if判断用户的roleId，若为普通用户则该按钮不予显示。第二个需求首先要在记录管理的vue中获取用户的权限id，并在后端添加if判断，若roldId为2（普通用户）则查询时仅能查询到同自己id的记录即可。至此，出入库权限控制和记录查询权限控制的开发就完成了。

4.3 主要功能测试

在第二章有提到目前最为主流的前后端分离开发模式，前端技术人员和后端人员基于接口文档开发前后端程序。但是对于后端人员来说，在开发程序时是没有前端页面的，那该如何测试开发的功能接口呢？我们当然可以直接使用浏览器，通过输入本机地址和端口号测试程序。但这种方式都是get请求，若碰到post请求还需程序员编写前端代码。因此，我们要使用专业的接口测试工具Postman，它是一款功能强大的网页调试与发送HTTP请求的Chrome插件。

例如在通过post请求JSON格式的参数时，我们便可以通过Postman工具进行调试，填写请求方式和请求路径便可发送请求。若响应结果符合预期，则在前端页面开发完毕时，可直接调用后端接口实现功能对接。

代码清单 4.6 JSON请求参数

|  |
| --- |
| @RequestMapping("jsonParam") public String jsonParam(@RequestBody User user) {  System.out.println(user);  return "OK"; } |

例如在通过post请求JSON格式的参数时，我们便可以通过Postman工具进行调试，填写请求方式和请求路径便可发送请求。若响应结果符合预期，则在前端页面开发完毕时，可直接调用后端接口实现功能对接。

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

图 4.3 JSON请求参数测试

在项目开发中，功能测试是非常重要的一个环节，它能够保证产品的质量和稳定性。具体来说，功能测试可以帮助我们发现并修复软件中的缺陷和问题，确保产品的功能和业务流程能够正常运作，同时也能够提高产品的可靠性和用户体验。在功能测试中，我们可以通过模拟用户的使用场景和业务操作，对软件进行全面、系统性的测试，以确保软件的功能和业务流程是否符合需求和规范。本小节仅展示了一个后端接收请求参数的测试，系统的每个后端功能开发完成后都需要经此步骤进行调试。

4.4 本章小结

本章主要介绍了开发环境的搭建以及主要功能模块的实现。在4.2中，我们介绍了后端项目的创建和配置，以及前端页面的搭建，包括导航伸缩、数据渲染、分页展示、增删改查等基本功能的实现。包括使用Vue CLI创建项目、引入ElementUI、页面布局的拆分、表单校验、安装axios处理跨域问题等。我们还介绍了登录验证和路由管理的实现，以及权限管理的基本思路。包括退出登录和个人中心功能的完善、动态路由的生成以及Vuex的应用等。在4.3中，我们介绍了功能测试对于项目开发的重要性。

总之，本章主要围绕着实现一个基本的后台管理系统，通过具体的实现步骤和详细的代码演示，为读者提供了一个从零开始构建项目的实践案例，帮助读者深入理解前后端开发的基本流程和技术要点。系统开发利用SpringBoot框架，采取分层思想，易于实现，便于扩展，系统功能完善，运行稳定[10]。通过本章的讲解，读者可以更好地了解Vue.js框架、Element-UI组件库等前端技术的使用，以及Spring、Node.js等后端技术的应用。

# 结 论

本文从对仓库管理系统的开发需求出发，采用基于B/S架构的前后端分离开发模式，通过SpringBoot、Mybatis、Vue.js和ElementUI等关键技术实现了开发。首先通过对用户需求、页面设计、系统关键功能、数据库、可行性等发面的分析，完成了对仓库管理系统的设计。在设计完成的基础上进行实际功能的开发，成功实现了系统的主要功能模块，包括页面布局的搭建、导航栏伸缩、数据渲染、分页展示、表单校验、数据的增删改查、登录验证、路由管理和权限管理等。

该系统采用了前后端分离开发模式，将前端与后端进行了彻底的分离，实现了数据与界面的解耦，提高了系统的可拓展性与可维护性。本文在实践中发现，使用SpringMVC三层框架开发Web应用时，应当合理设计每层的职责，避免分工不明确导致交叉重复等问题的出现，而增加系统的耦合性。在实际开发部署时，应当注重前端页面的渲染性能，采用如减少HTTP请求、启用压缩技术等手段进行优化，提高页面的加载速度，优化用户体验。

从实现结果来看，本文所提出的仓库管理系统能够基本满足用户的需求，实现了对仓库物品及用户管理的基本功能。但是，本系统除了在配置Mybatis-Plus的分页插件时用到了拦截器（Interceptor），未使用到过滤器等其他技术。过滤器在JavaWeb开发中可以通过拦截请求和响应对请求进行过滤和预处理。在前后端分离设计架构的系统中，过滤器可以起到很好的作用，例如对请求的合法性进行校验，过滤掉非法请求，避免数据被恶意攻击或非法注入等安全问题。

因此，在未来的研究和实践中，我们不仅要考虑系统的可维护性和可拓展性，更要考虑系统的安全性。同时，我们也要注重对JavaWeb开发中其他技术的学习及应用，不断优化和提升系统的性能和用户体验。

# 致 谢

时光荏苒，岁月如梭，大学四年的时光转眼就要过去。回想在学校经历的点点滴滴，不舍的心情难以言表。在此，我想感谢学校为我们提供了一个难得可贵的良好学习环境。

本人毕业设计的完成，离不开王子健老师的悉心指导和技术帮助。由衷感谢王老师及所有传授知识于我的各位老师们，是你们的帮助让我能够对所学知识进行归纳总结，掌握对本专业知识自学的能力。在今后的工作生活中，我也会秉持虚心学习的态度，努力在互联网行业发光发热。

与此同时，我也要感谢所有从事相关领域研究的业界前辈，因为有你们分享的研究成果供我以参考学习，我才能够完成这篇论文的部分理论空缺。感谢网络上分享相关技术教程的各位视频制作者，正是你们的无私奉献，才使在完成项目的路上的我不会走很多弯路。感谢在大学四年生活中帮助过、关心过我的人，感谢殷勤付出的老师和热情友善的学长学姐们，也感谢学校在疫情期间对我们的照料。

最后，我要感谢我的家人，是你们在背后默默支持着我，尊重我的选择。有你们的支持，我才能够在充满迷茫的年纪找到自己的方向，并顺利完成自己的学业。在今后的人生中，我会努力实现自己的人生价值，再次感谢你们。

# 参考文献

1. 鲍姗, 孟雅洁. 对基于SpringBoot微服务架构下的MVC模型分析[J]. 信息周刊, 2019(37): 100.
2. 朱蓉, 郑建华. 基于SpringBoot的垃圾分类科普及课程平台[J]. 电脑知识与技术, 2022, 18(09): 023.
3. 柳欢. 基于SpringBoot和Maybati-plus的信息发布管理系统[J]. 数码设计, 2021, (06): 032.
4. 关海伟. 基于JavaWeb的云存储技术研究[D]. 西安: 西安电子科技大学硕士学位论文, 2014.
5. 李峰, 刘彦隆. 基于SSH框架与jQuery技术的JavaWeb开发应用[J]. 科技情报开发与经济, 2010, 20(06): 106.
6. 戴亚峰. 基于JavaWeb的住宅小区物业管理系统设计与实现[D]. 江西: 江西财经大学硕士学位论文, 2018.
7. 边卫国. 基于MVC模式的JavaWeb框架研究与应用[D]. 西安: 西安电子科技大学硕士学位论文, 2006.
8. 朱炼, 左一博, 李如平. 快递终端管理系统设计与开发[J]. 榆林学院, 2021, 31(02): 061.
9. 刘柠新. 基于JavaWeb人力资源管理系统的设计与实现[D]. 广东: 广东工业大学硕士学位论文, 2019.
10. 刘超慧, 杨雨涵, 邢丹阳, 解秋寒, 李舶永. 基于SpringBoot的教学资源平台设计与实现[J]. 科技风, 2021, (11):092.

# 附 录

译文1

**KEY TECHNOLOGY**

System architecture:

C/S mode (Client / Server) and B/S mode (Browser / Server) are a mode that work together in current Internet applications. Due to the rise of Web browsers, B/S gradually replaced C/S and was more widely used. With the maturity and popularization of computer network technology, especially the development of local area networks and the emergence of PCs, more and more users and enterprises have begun to use computers to manage some transactions. B/S maintenance and upgrade are simple. As long as the server is operated, the client does not need to be changed.

Users only need to access the browser to access and operate, which simplifies the development, maintenance and use of the system. In the Belt and Road Business Cloud Service Platform system, the system uses the B/S structure for development. This system is the development of dynamic applications. It interacts with databases and caches. Fig.1. is the architecture of the system. After the system goes online, it needs to consider the large-scale concurrent interaction of users in countries along the Belt and Road. Multiple users log in to the system for use. Have a reliable, secure, and easy-to-maintain application.

Web Services Technology:

There are many server software released using WEB sites. Commonly used are Nginx, Sun, Jetty, Apache, etc. Most of the software can be used across platforms. However, it should be noted that due to the different architectures and operating principles of various software, various software may show different performance on different platforms. Functionality and operational performance are two key points to consider simultaneously for Web servers. This project is based on Tomcat server and Nodejs server for front-end separation.

Tomcat is an open source, small, lightweight application backend server. It is widely used in small and medium-sized systems and where there are not many concurrent users. Compared with the Apache server, Tomcat can dynamically generate resources and return them to the client. The dynamic resource is the opposite. The content accessed by different clients at different times is different, for example: a page containing the current time and a page showing the current IP address. Similar servers such as Apache and Nginx do not support generating dynamic pages, they must be supported by other modules.

Node.js is currently the fastest JavaScript engine with non-blocking, event-driven I/O and other features. It is more efficient than PHP and has a lower learning curve. The most important feature of Node.js is event-driven, which means that the server responds only when the user has an event. Node.js acts as a server for the browser to forward data to the back-end server and control the logic of page jumps with methods designed for routing.

Server technology:

The Spring open source framework is a research and development software product based on the JEE framework with the highest market share. The Spring MVC framework (MVC is the Model View Controller Organize code in a method that separates business logic, data, and interface display) launched by it is widely used in the research and development of various web products. Spring has released multiple versions, but the biggest difference between each version is that the tedious configuration work is gradually reduced as the version is updated, that is, the Spring framework has been committed to solving the tedious configuration work problem. The SpringBoot project is a new Spring framework developed by the Pivotal team. The purpose of its development is to solve the complex configuration operations in the Spring development process and ultimately achieve zero configuration. Its simplified application development and construction process frees developers from cumbersome configuration files and devotes a lot of energy to project logic research and development. It is committed to becoming a leader in rapid application development.

Database technology:

In the database management system (DBMS), there are many database products such as NoSql, Oracle, DB2, Mysql, etc. We choose to use the Mysql database according to the application support of interactive languages, database performance analysis, portability and other aspects. Mysql is a highly extensible database language that supports compilation on different compilers and works on different platforms. Provides C, Python, Java, PHP and other interfaces. Many large and small enterprises, websites and databases now use Mysql as the back-end database. Mysql will automatically balance and normalize to optimize performance. The database is responsible for ensuring the internal and external security of the database content. Mysql can configure authorized users to protect data to only be accessed by authorized users.

Crawler technology:

Web crawlers are also called web spiders, and their function is to download and analyze different web pages. Search engines such as Baidu and Google rely on a large number of web crawlers to search and analyze on the Internet, establish links to keywords in web pages, store them, calculate relevance, and sort them before displaying keywords to users.

At present, there are many mature and mature crawler frameworks that can be called through Python, such as Mechanize, BeautifulSoup, pyquery, Scrapy, etc. These libraries or frameworks have their own advantages, and Scrapy is one of the most prominent frameworks. Scrapy is a Python-based crawler. The Scrapy framework was developed to complete some structured data acquisition and other functions. Scrapy can be applied to a variety of common areas in daily life.

This system is a WEB management system. The frontend display uses CSS, Vue and other technologies to optimize the interface. The back-end uses python language to crawl the data and process the page data. The database uses MYSQL technology to store the data. The project contains seven functions. The first is the display of basic data. This block function can display the data of the database through charts and other forms, so that users can view and analyze the data as soon as they enter the system. The function of the investment management agency can Investment management personnel or institutions to perform related operations, or get in touch; etc. . The advisory function will display the crawled real-time latest information to users, making it easy for users to understand the latest global information. The detailed data module provides more detailed data references. At the same time, users can query the data, and can find out the trade data of multiple countries by macro, region, and time. Planning solutions help users plan data processing solutions. International business policies and regulations show international business policies and regulations.

中文译文

关键技术

系统架构：

C/S模式（客户端/服务器）和B/S模式（浏览器/服务器）是当前互联网应用程序中共同使用的一种模式。由于Web浏览器的兴起，B/S逐渐取代了C/S并被更广泛地使用。随着计算机网络技术的成熟和普及，尤其是局域网的发展和个人电脑的出现，越来越多的用户和企业开始使用计算机管理一些交易。B/S的维护和升级简单。只要服务器在运行，客户端就无需更改。用户只需访问浏览器即可访问和操作，这简化了系统的开发、维护和使用。在“一带一路”商务云服务平台系统中，系统采用了B/S结构进行开发。该系统是动态应用程序的开发，它与数据库和缓存进行交互。图1是系统的架构图。系统上线后，需要考虑沿着“一带一路”国家的大规模并发用户交互。多个用户登录系统进行使用。需要有一个可靠、安全、易于维护的应用程序。

Web服务技术：

有许多使用WEB站点发布的服务器软件。常用的有Nginx、Sun、Jetty、Apache等。大多数软件可跨平台使用。但需要注意的是，由于各种软件的不同架构和操作原理，各种软件在不同平台上可能会显示不同的性能。功能性和操作性能是同时考虑的两个关键点。这个项目基于Tomcat服务器和Nodejs服务器进行前端分离。

Tomcat是一种开源、小型、轻量级应用后端服务器。它广泛用于小型和中型系统以及并发用户不多的地方。与Apache服务器相比，Tomcat可以动态生成资源并将其返回给客户端。动态资源是相反的。不同客户端在不同时间访问的内容是不同的，例如：包含当前时间的页面和显示当前IP地址的页面。类似的服务器如Apache和Nginx不支持生成动态页面，它们必须由其他模块支持。

Node.js目前是最快的JavaScript引擎，具有非阻塞、事件驱动I/O等特性。它比PHP更高效，学习曲线更低。Node.js最重要的特点是事件驱动，这意味着只有用户有事件时服务器才会响应。Node.js充当浏览器的服务器，将数据转发到后端服务器，并使用专门用于路由的方法控制页面跳转的逻辑。

服务器技术：

Spring开源框架是基于JEE框架的一种研究和开发软件产品，市场占有率最高。由Spring推出的Spring MVC框架（MVC是模型-视图-控制器模式，将业务逻辑、数据和界面展示分开编码）被广泛应用于各种Web产品的研发中。Spring发布了多个版本，但每个版本之间最大的区别在于，随着版本的更新，繁琐的配置工作逐渐减少，也就是说，Spring框架一直致力于解决繁琐的配置工作问题。SpringBoot项目是由Pivotal团队开发的新型Spring框架。其开发目的是解决Spring开发过程中的复杂配置操作，最终实现零配置。它简化了应用程序的开发和构建过程，使开发人员摆脱了繁琐的配置文件，将大量精力投入到项目逻辑研究和开发中。它致力于成为快速应用程序开发的领导者。

数据库技术：

在数据库管理系统（DBMS）中，有许多数据库产品，如NoSql、Oracle、DB2、Mysql等。我们选择使用Mysql数据库，根据交互式语言的应用支持、数据库性能分析、可移植性等方面。Mysql是一种高度可扩展的数据库语言，支持在不同编译器上进行编译，并在不同平台上工作。提供C、Python、Java、PHP等接口。许多大型和小型企业、网站和数据库现在都使用Mysql作为后端数据库。Mysql会自动平衡和规范化以优化性能。数据库负责确保数据库内容的内部和外部安全。Mysql可以配置授权用户，保护数据只能被授权用户访问。

爬虫技术：

Web爬虫也称为网络蜘蛛，其功能是下载和分析不同的网页。百度和Google等搜索引擎依赖于大量的Web爬虫在互联网上搜索和分析，建立关键词与网页之间的链接，存储并计算相关性，然后在向用户显示关键词之前对它们进行排序。

目前，有许多成熟的爬虫框架可以通过Python进行调用，例如Mechanize，BeautifulSoup，pyquery，Scrapy等。这些库或框架都有自己的优点，而Scrapy是最出色的框架之一。Scrapy是基于Python的爬虫。Scrapy框架的开发是为了完成一些结构化数据采集和其他功能。Scrapy可以应用于日常生活中的各种常见领域。

该系统是一个WEB管理系统。前端展示使用CSS、Vue等技术来优化界面。后端使用Python语言爬取数据并处理页面数据。数据库使用MYSQL技术来存储数据。该项目包含七个功能。第一个是基本数据的显示。该功能可以通过图表等形式显示数据库的数据，使用户在进入系统后可以快速查看和分析数据。投资管理机构的功能可以让投资管理人员或机构执行相关操作，或者联系等等。咨询功能将向用户显示爬取的实时最新信息，使用户易于了解最新的全球信息。详细数据模块提供更详细的数据参考。同时，用户可以查询数据，并可以通过宏观、地区和时间了解多个国家的贸易数据。规划解决方案帮助用户规划数据处理方案。国际商业政策和法规显示国际商业政策和法规。

译文2

**Programming languages in chemistry: a review of HTML5/JavaScript**

Computer literacy is an essential skill for scientists. Computers can execute the same operations humans can perform, but far more consistently and at rates far beyond human capability, allowing researchers to investigate numerous hypotheses in short order. Programmers have even more advantages, as they can directly communicate with the computer to achieve their desired goals, as opposed to relying on software someone else has created for another purpose.

Many programming languages have been developed to facilitate instructions to the computer. Each has its advantages, which is why they each exist. Each programming language also has its disadvantages, which is why the rest exist.

When the Internet was introduced, society quickly changed, not just in allowing computers to communicate with each other, but by allowing people to communicate with each other, nearly instantaneously from around the world. JavaScript (JS) is a programming language born of the Internet. From a crude and basic programming language for creating cheap, gimmicky effects on web pages, it has developed into a ubiquitous and flexible technology where engine performance is regarded as a crowning achievement among browser developers. Today, every computational device, from desktops to laptops to mobile phones and Augmented Reality (AR) devices has essential access to the Internet, and therefore contains a web browser wherein JavaScript can be run. As a result, JavaScript is one of the most important programming languages in existence.

This review investigates the relevance and impact of JavaScript on the chemistry discipline. We begin by discussing the history of the programming language; then we investigate the scope and features of the language and how it applies to chemistry; last, an outline of the technical details of the JavaScript programming language is presented to allow those interested in chemistry and cheminformatics to effectively create JavaScript applications.

JavaScript is a programming language enabling developers to interact with the functionality provided by web browsers. More specifically, JavaScript is a scripting language, which means traditionally, JavaScript source code is interpreted at runtime and not pre-compiled into byte code and practically, its main purpose is to modify the behavior of another application typically written in a different programming language, in which it is interpreted and run in real time.

While JavaScript is aptly named as a scripting language, the first part of the name misleadingly refers to the Java programming language. JavaScript has no functional relationship with the Java programming language. The Java part of JavaScript was used to inspire interest in the new scripting language, as Java was and still is a very popular programming language; additionally the original Netscape browser that included JavaScript was written in Java. JavaScript has since become an essential and arguably more important programming language than the one inspiring its name. Hence, the programming language many developers use today to build and interact with web applications is famously known as JavaScript.

In the early days of the Internet, the web browser provided users with static pages to view information. Innovators at Netscape Communications, producers of the popular Netscape Navigator browser, believed web sites should provide more dynamic material and a programming language would provide a solution. In 1995, Brendan Eich, an employee of Netscape Communications, developed what is now known as JavaScript.

Initially, JavaScript only provided basic features, but the power of such a programming language on the web was quickly realized, and the continued success of Netscape Navigator was, in no small part, a reflection of that power. Alternatives to JavaScript were inevitable and a number of companies started to produce more feature-rich browser plug-ins that would install their platforms into browsers to be run in web pages. Adobe Flash, Java applets and Microsoft Silverlight are a few well known examples. These plug-ins became popular solutions for developers, but a hindrance for users as plug-ins required installations, had to be frequently updated, and were prone to security issues. Plug-ins eventually fell out of favor with users as HTML5 arrived and JavaScript evolved. Flash, Java applets and Silverlight are all now deprecated technologies.

Other companies, instead of replacing JavaScript, attempted to modify and extend the language in an attempt to control it (a process known as “embrace, extend and extinguish”). In 1996, Microsoft produced JScript, which was a reverse engineered version of JavaScript for their Internet Explorer (IE) browser. IE would now be accessible to JavaScript developers, but the implementation was deceitful. Proprietary features specific to IE were built into JScript. As a result, users were forced to use IE for many online applications, impeding the open web in a process known as fragmentation. Fragmentation produces an anti-competitive, positive feedback loop giving a dominant group (in this case, the dominant browser, IE) power over the future direction of the Internet. These attempts ultimately failed due to healthy competition amongst browser developers. Microsoft browsers now aim to legitimately support JavaScript. Creative methods for fragmenting the JavaScript language continue to exist with TypeScript from Microsoft and Dart from Google, which aim to provide a more familiar programming syntax for developers and compile into standard JavaScript.

JavaScript has survived many challenges since its inception, and a lot of credit should be given to the Netscape team for their foresight. In order to protect the JavaScript language early on, the Netscape team pushed for an open standardization of the language, and they were successful. We all benefit from the universal and strong support for JavaScript in every existing browser today. In spite of the attempts to replace or commandeer the standard, JavaScript persisted and has become a signature programming language for most developers, and remains an important tool for the scientific communities including chemists.

JavaScript is far from the end of its life, rather it seems like it is just beginning. While JavaScript exhibits many weaknesses, its strengths are paramount, allowing not just the creation of pedestrian web based solutions, but in specifically providing a means for communicating the complexity of chemistry. As an open and standardized language, JavaScript has continued to thrive and evolve, while remaining a reliable foundation for developers. Scientists continue to find better and more powerful ways to use web technologies in the pursuit of science and to make knowledge accessible around the world. We will undoubtedly see continued technological innovation, and JavaScript, as the internet browser programming language, will likely continue to be the tool of choice for web developers and essential for the propagation of scientific information.

中文译文

化学中的编程语言：HTML5/JavaScript综述

计算机素养对科学家来说是一项必不可少的技能。计算机能够执行与人类相同的操作，但远比人类能力更一致，且速率也快得多，让研究人员能够在短时间内探究众多假设。程序员更有优势，因为他们可以直接与计算机通信，实现自己想要的目标，而不是依靠他人为另一个目的创建的软件。

许多编程语言已经被开发出来，以便向计算机发出指令。每种语言都有其优势，这就是它们各自存在的原因。每种编程语言也都有其不足之处，这就是其他语言存在的原因。

当互联网被引入后，社会迅速发生了变化，不仅让计算机相互通信，还让人们能够几乎瞬间与世界各地的人们通信。JavaScript（JS）是一种由互联网诞生的编程语言。它最初是用于在网页上创建廉价、花哨的效果的粗糙基础编程语言，但已经发展成为一种无处不在且灵活的技术，其中引擎性能被视为浏览器开发者的巅峰成就。今天，从台式机到笔记本电脑再到手机和增强现实（AR）设备，每一种计算设备都必须要连接互联网，因此都包含能够运行JavaScript的网络浏览器。因此，JavaScript是现存最重要的编程语言之一。

本综述旨在研究JavaScript对化学学科的相关性和影响。我们首先讨论了这种编程语言的历史；然后我们调查了这种语言的范围和特点以及它在化学中的应用；最后，我们提供了JavaScript编程语言的技术细节概述，以便对化学和化学信息学感兴趣的人有效地创建JavaScript应用程序。

JavaScript是一种编程语言，使开发人员能够与Web浏览器提供的功能进行交互。更具体地说，JavaScript是一种脚本语言，这意味着传统上，JavaScript源代码在运行时被解释，而不是预编译成字节码，并且其主要目的是修改通常由不同编程语言编写的另一个应用程序的行为，在该应用程序中它被解释和实时运行。

虽然JavaScript被恰当地称为脚本语言，但其名称中的前半部分误导性地指的是Java编程语言。JavaScript与Java编程语言没有功能上的关系。JavaScript中的Java部分是用来激发对这种新脚本语言的兴趣的，因为Java是当时非常流行的编程语言，此外最初包括JavaScript的Netscape浏览器是用Java编写的。此后，JavaScript已成为一种重要的、比它的名字所示更重要的编程语言。因此，如今许多开发人员用于构建和与Web应用程序交互的编程语言都以JavaScript著名。

在互联网的早期，Web浏览器向用户提供静态页面来查看信息。Netscape通讯公司的创新者认为，网站应提供更多动态内容，编程语言将提供解决方案。1995年，Netscape通讯公司的员工Brendan Eich开发出现在的JavaScript。

在互联网的早期，网页浏览器只提供用户查看静态页面的功能。Netscape Communications是生产流行的Netscape Navigator浏览器的公司，他们认为网站应该提供更多动态材料，并认为编程语言是解决方案。1995年，Netscape Communications的员工Brendan Eich开发了现在称为JavaScript的语言。

起初，JavaScript只提供了基本功能，但网页上这种编程语言的强大之处很快被认识到，而Netscape Navigator的持续成功在很大程度上反映了这种力量。JavaScript的替代方案是不可避免的，许多公司开始生产更具特色的浏览器插件，将其平台安装到浏览器中以在网页中运行。Adobe Flash、Java applets和Microsoft Silverlight是几个知名的例子。这些插件成为开发人员的流行解决方案，但对用户来说是个障碍，因为插件需要安装，必须经常更新，并容易存在安全问题。随着HTML5的出现和JavaScript的发展，插件最终不再受用户青睐，Flash、Java applets和Silverlight都是现已弃用的技术。

其他公司并没有替换JavaScript，而是尝试修改和扩展这种语言，以试图控制它（这个过程称为“拥抱、扩展和扼杀”）。1996年，Microsoft推出了JScript，这是他们的Internet Explorer（IE）浏览器的JavaScript的反向工程版本。现在，IE可以被JavaScript开发人员访问，但实现方式是欺骗性的。专有功能特定于IE的功能被内置到JScript中。因此，用户被迫在许多在线应用程序中使用IE，阻碍了开放网络，这个过程称为分裂。分裂产生了一种反竞争的正反馈循环，使得占主导地位的群体（在本例中，是占主导地位的浏览器IE）掌握了互联网未来方向的权力。这些尝试最终由于浏览器开发人员之间的健康竞争而失败。现在，Microsoft浏览器旨在合法地支持JavaScript。通过TypeScript和Google的Dart等创造性的分裂JavaScript语言的方法仍然存在，这些方法旨在为开发人员提供更熟悉的编程语法，并编译为标准的JavaScript。

自JavaScript问世以来，它已经经历了许多挑战，Netscape团队应该得到很多赞誉。为了保护JavaScript语言，Netscape团队早期推动了对该语言的开放标准化，并取得了成功。今天，我们所有人都从每个现有浏览器JavaScript的普遍和强大支持中受益。尽管有试图替换或掌控标准的尝试，JavaScript仍然存在，并已成为大多数开发人员的标志性编程语言，对包括化学家在内的科学界来说仍然是一个重要工具。

JavaScript远未走到生命的尽头，相反，它似乎才刚刚开始。尽管JavaScript存在许多弱点，但其优点至关重要，不仅可以创建普通的基于Web的解决方案，而且在特定情况下为传达化学复杂性提供手段。作为一种开放和标准化的语言，JavaScript一直在茁壮成长和发展，同时仍然是开发人员可靠的基础。科学家继续发现更好更强大的使用Web技术追求科学和使知识普及全球的方式。我们毫无疑问将看到持续的技术创新，而作为互联网浏览器编程语言的JavaScript很可能会继续成为Web开发人员的首选工具，并对科学信息的传播至关重要。