

南通大學



本科毕业设计

题目	基于 Node.js 的 Web 框架生成工具
----	-------------------------

作者：_____ 韦远 _____

专业：_____ 计算机科学与技术 _____

指导教师：_____ 朱晓辉 _____

完成日期：_____ 2017 年 5 月 31 日 _____

原创性声明

本人声明：所呈交的论文是本人在导师指导下进行的研究成果。除了文中特别加以标注和致谢的地方外，论文中不包含其他人已发表或撰写过的研究成果。参与同一工作的其他同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

签 名：_____日 期：_____

本论文使用授权说明

本人完全了解南通大学有关保留、使用学位论文的规定，即：学校有权保留论文及送交论文复印件，允许论文被查阅和借阅；学校可以公布论文的全部或部分内容。

（保密的论文在解密后应遵守此规定）

学生签名：____指导教师签名：____日期：_____

南通大学毕业设计（论文）立题卡

课题名称	基于 Node.js 的 Web 框架生成工具		出题人	朱晓辉
课题表述 (简述课题的背景、目的、意义、主要内容、完成课题的条件、成果形式等)	<p>课题背景： 随着 web 开发技术的不断更新增长，web 开发越来越多样化，复杂化</p> <p>意义： 提供一套 web 开发的基本模板管理 web 开发的各个模块，让开发更加快速高效，管理更加便捷</p> <p>主要内容：</p> <p>（1）一套基于 Node.js 的 web 模版</p> <p>（2）展示模版外观，功能，特性的 web 网站</p> <p>（3）可供用户的选择不同的样式</p> <p>成果形式： 一套软件，毕业论文一篇</p>			
课题来源	社会生产实际	课题类别	毕业设计	
该课题对学生的要求	<p>要求学生熟悉 web 开发的 html，css，js 基本技术以及 Node.js 的开发框架 express 构建工具 Webpack 和 web 前端的开发框架 AngularJs，数据库 MongoDB</p>			
系、部意见	系、部主任签名： _____ _____年____月____日			
学院意见	同意立题（ ） 不同意立题（ ） 教学院长签名： _____ _____年____月____日			

注： 1、课题来源是指： 1.科研， 2.社会生产实际， 3. 其他。
 2、课题类别是指： 1.毕业论文， 2.毕业设计。

南 通 大 学

毕业设计（论文）任务书

题目 基于 Node.js 的 Web 框架生成工具

姓 名 韦远

学 院 计算机科学与技术学院

专 业 计算机科学与技术

班 级 计 132

学 号 1313022048

起讫日期 2017 年 1 月 12 日~2017 年 6 月 12 日

指导教师 朱晓辉 职称 副教授

发任务书日期 2017 年 1 月 12 日

<p>课题的内容和要求（研究内容、研究目标和解决的关键问题）</p>
<p>研究内容：（1）展示模版的站点的开发（2）模版的开发和模块之间的管理 （3）模版样式的开发</p> <p>研究目标：实现 web 模板的构建，用来更好的管理 web 开发的各个模块</p> <p>关键问题：（1）如何管理开发的各个模块（2）如何确保模版的可用性 （3）模版样式如何制作</p>
<p>课题的研究方法和技术路线</p>
<p>研究方法：</p> <p>根据用户对样式的选择完成 web 模板的开发</p> <p>技术路线：</p> <p>（1）通过 angularjs 完成前端框架开发</p> <p>（2）通过 nodejs 的 express 完成后端框架开发</p> <p>（3）通过 Webpack 对各个模块进行管理</p> <p>（4）通过 CSS 技术实现模板样式。</p> <p>（5）通过 nosql 的 mongodb 对数据进行管理</p>
<p>基 础 条 件</p>
<p>（1）充分熟悉使用者的需求。</p> <p>（2）熟悉 HTML、CSS、JavaScript、angularjs 等前端开发技术</p> <p>（3）熟悉 nodejs 的 express 框架和 grunt 构建工具</p> <p>（4）熟悉 nosql 的 mongodb</p>

参 考 文 献

- [1] 黄扬子 基于 NodeJS 平台搭建 REST 风格 Web 服务 无线互联科技 2015-08-25
- [2] 许会元; 何利力 NodeJS 的异步非阻塞 I/O 研究 工业控制计算机 2015-03-25
- [3] 柯肇丰; 曾霞霞 基于 HTML5+nodeJS+MongoDB 构建在线图像编辑器系统 福建电脑 2015-06-25
- [4] 郑立国; 于宏伟 基于 Nodejs 的本机应用与 Web 应用统一编程模型的研究 优先出版 电子技术与软件工程 2015-06-16
- [5] 郑杰 AngularJS 前端框架与后端代码自动生成系统的设计与实现 西南交通大学 2017-05-10
- [6] 冯晨超 基于 AngularJS 的物品和用户查询模块的设计与实现 南京大学 2015-05-01
- [7] 张文盛; 郑汉华 基于 MongoDB 构建高性能网站技术研究 吉林师范大学学报(自然科学版) 2013-02-10
- [8] 黄悦深 基于 HTML5 的移动 Web App 开发 图书馆杂志 2014-07-15
- [9] 杨俊; 李艳梅 JavaScript 面向对象编程探析 办公自动化 2010-04-15
- [10] 张俊兰; 刘翼; 铁宏军 基于 CSS 技术的网页设计应用研究 延安大学学报(自然科学版) 2010-09-20
- [11] 周玲余 基于 jQuery 框架的页面前端特效的设计与实现 计算机与现代化 2013-01-15
- [12] Seonghoon Kang; Sukyoung Ryu ACM SIGPLAN Notices, 2012, Vol.47 (10), pp.621-638 Association for Computing Machinery Journal
- [13] Lubbers P, Albers B, Salim F Pro HTML5 Programming: Powerful APIs for Richer Internet Application Development 2010
- [14] Tilkov, Stefan, Vinoski, Steve. Node.js: Using JavaScript to Build High-Performance Network Programs[J]. IEEE Internet Computing . 2010 (6)
- [15] Erasmus M. .CoffeeScript Programming with JQuery, Rails, and Node. js. 2012
- [16] Rauch G. Smashing Node. js: JavaScript Everywhere. 2012

本课题必须完成的任务
<p>本课题必须完成的任务：</p> <p>（1）web 模板的开发：开发出一个对 web 开发的各个模块管理的统一的基本的模板。</p> <p>（2）web 网站的开发：根据用户的样式选择生成对应的 web 模板，并进行下载。</p> <p>（3）用户样式：开发出一套网站样式的模板。</p>
成 果 形 式
<ol style="list-style-type: none">1. 完成基于 Node.js 的 Web 框架生成工具项目的编写及运行2. 完成一篇关于该课题的毕业论文

进度计划		
起讫日期	工作内容	备 注
2017.1.6-2017.3.25	毕业设计调研；查阅资料；技术准备	
2017.3.26-2017.4.20	收集资料；完成毕业设计开题报告	
2017.4.21~2017.5.11	毕业设计编码	
2017.5.11~2017.5.21	编写、修改毕业论文	
2017.5.22~2017.6.12	准备毕业设计答辩	
系、部审核意见	<div></div> <div>系、部主任签名：_____ _____年__月__日</div>	
学院意见	<div></div> <div>教学院长签名：_____ _____年__月__日</div>	

南通大学本科生毕业设计（论文）开题报告

学生姓名	韦远	学 号	13130220 48	专业	计算机科学与技术
课题名称	基于 Node.js 的 Web 框架生成工具				
阅读文献 情 况	国内文献 11 篇	开题日期	2017 年 3 月 16 日		
	国外文献 5 篇	开题地点	JSJ402		
<p>一、 文献综述与调研报告：（阐述课题研究的现状及发展趋势，本课题研究的意义和价值、参考文献）</p> <p>现状及发展趋势：</p> <p>随着 Web 开发技术的不断更新增长，Web 开发越来越多样化，复杂化，现如今，Web 开发技术种类繁多，很多人在进行 Web 开发时还不知道如何选择开发技术以及不知道选择何种开发技术，有的甚至习惯于一层不变的使用一些已经落后的被淘汰的技术，从而使得 Web 开发变得低效而且复杂化。</p> <p>意义和价值：</p> <p>首先可以让用户了解到一些 Web 开发的常用技术，然后根据项目的业务需求选择相应的技术生成 Web 模版，最后使用该模版来开发网站。从而实现 Web 开发的快速，高效，模块化等功能。</p> <p>参考文献：</p> <p>[1] 黄扬子 基于 NodeJS 平台搭建 REST 风格 Web 服务 无线互联科技 2015-08-25</p> <p>[2] 许会元; 何利力 NodeJS 的异步非阻塞 I/O 研究 工业控制计算机 2015-03-25</p> <p>[3] 柯肇丰; 曾霞霞 基于 HTML5+nodeJS+MongoDB 构建在线图像编辑器系统 福建电脑 2015-06-25</p> <p>[4] 郑立国; 于宏伟 基于 Nodejs 的本机应用与 Web 应用统一编程模型的研究 优先出版电子技术与软件工程 2015-06-16</p> <p>[5] 郑杰 AngularJS 前端框架与后端代码自动生成系统的设计与实现 西南交通大学 2017-05-10</p> <p>[6] 冯晨超 基于 AngularJS 的物品和用户查询模块的设计与实现 南京大学 2015-05-01</p>					

- [7] 张文盛; 郑汉华 基于 MongoDB 构建高性能网站技术研究 吉林师范大学学报(自然科学版) 2013-02-10
- [8] 黄悦深 基于 HTML5 的移动 Web App 开发 图书馆杂志 2014-07-15
- [9] 杨俊; 李艳梅 JavaScript 面向对象编程探析 办公自动化 2010-04-15
- [10] 张俊兰; 刘翼; 铁宏军 基于 CSS 技术的网页设计应用研究 延安大学学报(自然科学版) 2010-09-20
- [11] 周玲余 基于 jQuery 框架的页面前端特效的设计与实现 计算机与现代化 2013-01-15
- [12] Seonghoon Kang; Sukyoung Ryu ACM SIGPLAN Notices, 2012, Vol.47 (10), pp.621-638
Association for Computing Machinery Journal
- [13] Lubbers P, Albers B, Salim F Pro HTML5 Programming: Powerful APIs for Richer Internet
Application Development 2010
- [14] Tilkov, Stefan, Vinoski, Steve. Node.js: Using JavaScript to Build High-Performance Network
Programs[J]. IEEE Internet Computing . 2010 (6)
- [15] Erasmus M. .CoffeeScript Programming with JQuery, Rails, and Node. js. 2012
- [16] Rauch G. Smashing Node. js: JavaScript Everywhere. 2012

二、 本课题的基本内容， 预计解决的难题

基本内容：

实现 web 模板的构建，实现用户样式的选择，实现提供用户下载 web 模板的网站

预计解决的难题：

- (1) 如何管理开发的各个模块
- (2) 如何确保模版的可用性
- (3) 模版样式如何制作

三、 课题的研究方法、技术路线

研究方法：

根据用户对样式的选择完成 web 模板的开发

技术路线：

- (1) 通过 angularjs 完成前端框架开发
- (2) 通过 nodejs 的 express 完成后端框架开发
- (3) 通过 Webpack 对各个模块进行管理
- (4) 通过 CSS 技术实现模板样式。
- (5) 通过 nosql 的 mongodb 对数据进行管理

四、 研究工作条件和基础

- (1) 充分熟悉使用者的需求。
- (2) 熟悉 HTML、CSS、JavaScript、angularjs 等前端开发技术
- (3) 熟悉 nodejs 后端开发技术
- (4) 熟悉 Webpack 构建工具
- (5) 熟悉 nosql 的 mongodb

五、 进度计划					
起讫日期		工作内容			
2017.1.6-2017.3.25		毕业设计调研；查阅资料；技术准备			
2017.3.26-2017.4.20		收集资料；完成毕业设计开题报告			
2017.4.21-2017.5.11		毕业设计编码			
2017.5.11-2017.5.21		编写、修改毕业论文			
2017.5.22-2017.6.12		准备毕业设计答辩			
论文阶段完成日期		文献调研完成日	2017.3.20	论文实验完成日	2017.5.11
		撰写论文完成日	2017.5.21	评议答辩完成日	2017.6.10
指导教师评语	<div>导师签名：_____ 年__月__日</div>				
系部意见	<div>系、部主任签名：_____ 年__月__日</div>				
学院意见	<div>通过开题（ ）</div> <div>开题不通过（ ）</div> <div>教学院长签名：_____ 年__月__日</div>				

南 通 大 学

毕 业 设 计

题 目 基于 Node.js 的 Web 框架生成工具

姓 名 韦远

学 院 计算机科学与技术学院

专 业 计算机科学与技术

班 级 计算机 132

学 号 1313022048

指导教师 朱晓辉 职称 副教授

南通大学计算机科学与技术学院

2017 年 5 月 31 日

摘要

随着互联网技术的不断发展，网站已经成为人们不可或缺的一部分，但是随着 Web 技术的不断发展，各式各样的 Web 前端与后端的技术出现在了市场上面，让程序员不知道该如何选择。所以为了网站快速而高效的开发，本课题创建了一个以 Angular 为前端框架，Node.js 为后端支撑，Mongodb 为数据支持的网站模板，本系统功能架构大致分为六个部分：关键技术介绍模块，选择技术生成框架模块，Web 框架生成模块，用户下载使用框架模块，系统扩展性设计模块，框架设计模块，再配合 Webpack 强大的打包功能，以及前后端统一使用 JavaScript 语言开发，使用我们的这个模板来开发网站将体会到前所未有的快速轻量与高效。

关键字：Web 框架，Node.js，Angular

ABSTRACT

With the continuous development of Internet technology, the site has become an indispensable part of people, but with the continuous development of Web technology, a wide range of Web front-end and back-end technology has appeared on the market, so that programmers do not know how to choose them. Therefore, for the rapid and efficient development of the website, this topic creates a website template with Angular as the front-end framework, Node.js as the back-end support, and MongoDB for the data support. The system architecture is divided into six parts: the key technology introduction module, the choice of technology to generate the framework module, the Web framework generation module, the user download and use module, the system expansion design module, the framework design module, together with the Webpack powerful packing function, as well as both front and back are using JavaScript, using our template to develop the site will experience unprecedented fast lightweight and efficient.

Keywords: Web framework, Node.js, Angular

目录

摘要	i
ABSTRACT	ii
第一章 系统概述	1
1.1 研究背景	1
1.2 研究意义	1
1.3 开发环境和技术介绍	1
第二章 系统需求分析	7
2.1 系统设计目标	7
2.2 系统功能分析	7
2.2.1 展现 Web 开发技术功能	7
2.2.2 选择技术生成框架功能	8
2.2.3 打包管理模块功能	8
2.3 系统可行性分析	9
2.4 研究方法与技术路线	9
第三章 系统概要设计	11
3.1 系统设计思路	11
3.2 系统功能架构	11
3.2.1 Web 网站架构	11
3.2.2 Web 框架架构	12
第四章 系统详细设计	17
4.1 关键技术介绍模块	17
4.2 选择技术生成框架模块	17
4.3 Web 框架生成模块	18
4.4 用户下载使用框架模块	19
4.5 系统扩展性设计模块	20
4.6 框架设计模块	20
4.6.1 框架前端设计	20
4.6.2 框架后端设计	21

第五章 系统实现23

 5.1 系统展示23

 5.2 系统编码25

第六章 结论30

 全文总结30

 展望30

参考文献31

致谢32

第一章 系统概述

1.1 研究背景

随着 Web 开发技术的不断更新增长，Web 开发越来越多样化，复杂化，现如今，Web 开发技术种类繁多，很多人在进行 Web 开发时还不知道如何选择开发技术以及不知道选择何种开发技术，有的甚至习惯于一层不变的使用一些已经落后的被淘汰的技术，从而使得 Web 开发变得低效而且复杂化。

1.2 研究意义

首先可以让用户了解到一些 Web 开发的常用技术，然后根据项目的业务需求选择相应的技术生成 Web 模版，最后使用该模版来开发网站。从而实现 Web 开发的快速，高效，模块化等功能。

1.3 开发环境和技术介绍

本系统的网站开发采用 AngularJs 和 Bootstrap 作为前端框架，使用 Webpack 作为前端的打包管理工具，后端则采用 Node.js 作为支撑，数据库采用 Nosql 的 Mongodb。

相关技术介绍：

1. Node.js

Node.js 是一个用于执行 JavaScript 代码服务器端的开放源码的跨平台 JavaScript 运行时环境。Node.js 使 JavaScript 可用于服务器端脚本，并在脚本发送到用户的 Web 浏览器之前运行脚本服务器端来生成动态网页内容。因此，Node.js 已经成为“JavaScript 无处不在”范例的基础元素之一，允许 Web 应用程序开发统一单一的编程语言，而不是依靠不同的语言编写服务器端脚本。

虽然.js 是 JavaScript 代码的常规文件扩展名，但是“Node.js”的名称不是指在此上下文中的特定文件 - 它只是产品的名称。Node.js 具有能够进行异步 I / O 的事件驱动架构。这些设计选择旨在通过许多输入/输出操作以及实时 Web 应用程序（例如实时通信程序和浏览器游戏）来优化 Web 应用程序的吞吐量和可扩展性。

Node.js 允许使用 JavaScript 和一系列处理各种核心功能的“模块”来创建 Web 服务器和网络工具。为文件系统 I/O，网络（DNS，HTTP，TCP，TLS / SSL 或 UDP），二进制数据（缓冲区），加密功能，数据流和其他核心功能提供模块。Node.js 的模块使用一个旨在降低编写服务器应用程序的复杂性的 API。

Node.js 主要用于构建网络程序，如 Web 服务器。Node.js 和 PHP 之间最大的区别是，PHP 中的大多数函数都将被阻塞直到完成（仅在之前的命令完成后才执行命令），而 Node.js 中的函数被设计为非阻塞（命令并行执行并使用回叫信号完成或失败）。

Node.js 为 Web 服务器提供事件驱动的编程，可以在 JavaScript 中开发快速的 Web 服务器。开发人员可以创建高度可扩展的服务器，而无需使用线程，通过使用事件驱动编程的简化模型，使用回调来表示完成任务。Node.js 的创建是因为许多服务器端编程语言中的并发性很难，而且往往导致性能下降。Node.js 将脚本语言（JavaScript）与 Unix 网络编程的强大功能相结合。

Node.js 具有如下的技术特点：

- （1）Node.js 是一个 Javascript 运行时环境，它处理一个循环中的传入请求，称为事件循环。
- （2）Node.js 在单线程上运行，使用非阻塞 I/O 调用，允许它支持数万个并发连接，而不会导致线程上下文切换的成本。在使用观察者模式的所有请求之间共享单线程的设计旨在构建高度并发的应用程序，其中执行 I/O 的任何功能都必须使用回调。为了适应单线程事件循环，Node.js 利用 libuv 库，该库依次使用固定大小的线程池，该线程池负责一些非阻塞异步 I/O 操作。这种单线程方法的缺点是，Node.js 不允许通过增加其运行的机器的 CPU 内核数量而不使用附加模块（如集群） StrongLoop Process Manager 或 pm2。但是，开发人员可以增加 libuv 线程池中的默认线程数；这些线程可能由服务器操作系统分布在多个核心上。在 Node.js 中执行并行任务由线程池处理。主线程调用函数将任务发布到线程池中的线程的共享任务队列中，并执行。固有的非阻塞系统功能（如网络）转换为内核侧非阻塞套接字，而固有地阻止系统功能（如文件 I/O）在其自身线程上以阻塞方式运行。当线程池中的线程完成任务时，它会通知主线程，该

线程又唤醒并执行已注册的回调。由于回调在主线程上串行处理，持久的计算和其他 CPU 绑定的任务将冻结整个事件循环直到完成。

- (3) V8 是为 Google Chrome 构建的 JavaScript 执行引擎，由 Google 于 2008 年开放。V8 编写 C ++，将 JavaScript 源代码编译为本机代码，而不是实时解释。
- (4) Node.js 使用 libuv 处理异步事件。Libuv 是基于 Windows 和 POSIX 的系统（如 NonStop 和 Unix 上的 Linux，macOS，OSS）的网络和文件系统功能的抽象层。
- (5) Node.js 的核心功能驻留在 JavaScript 库中。以 C ++编写的 Node.js 绑定将这些技术相互连接并连接到操作系统。
- (6) npm 是 Node.js 服务器平台的预装包管理器。它用于从 npm 注册表安装 Node.js 程序，组织第三方 Node.js 程序的安装和管理。不要将 npm 与 CommonJS require（）语句混淆。它不用于加载代码；而是用于从命令行安装代码并管理代码依赖关系。在 npm 注册表中找到的软件包可以从简单的帮助程序库（如 Underscore.js）到任务运行程序（如 Grunt）。
- (7) Node.js 可以与浏览器，支持 JSON 数据的数据库（如 Postgres， MongoDB 或 CouchDB）和 JSON 结合使用，用于统一的 JavaScript 开发堆栈。通过适应本质上是 MVC，MVP，MVVM 等服务器端开发模式，Node.js 允许在客户端和服务端之间重用相同的模型和服务接口。
- (8) Node.js 向操作系统注册，以便在连接时通知，并且操作系统将发出回调。在 Node.js 运行时，每个连接都是一个小的堆分配。传统上，相对重量级的操作系统进程或线程处理每个连接。与其他事件驱动的服务器相反，Node.js 的事件循环不需要显式调用。而是定义了回调函数，服务器会自动在回调定义的末尾输入事件循环。当没有进一步的回调时，Node.js 退出事件循环。

2. AngularJs

AngularJS 是基于 JavaScript 的开源前端 Web 应用程序框架，主要由 Google 和个人和企业社区维护，以解决许多开发单页应用程序遇到的挑战。JavaScript 组件补充了 Apache Cordova，用于开发跨平台移动应用程序的框架。它旨在通过为客户端模型视图 - 控制器（MVC）和模型视图模型（MVVM）架构以及通常在富 Internet 应用程序中使用的组件提

供框架来简化开发和测试这些应用程序。2014 年，原来的 AngularJS 团队开始在 Angular（应用平台）上工作。

AngularJS 框架通过首先阅读 HTML 页面，其中嵌入了其他自定义标签属性。Angular 将这些属性解释为将页面的输入或输出部分绑定到由标准 JavaScript 变量表示的模型的指令。根据 JavaScript 分析服务 Libscore，AngularJS 被用于 Wolfram Alpha, NBC, Walgreens, Intel, Sprint, ABC News 以及 2016 年 10 月测试的 100 万个网站中的大约 12,000 个其他网站。AngularJS 是 GitHub 上所有时间最多的第六大项目。

AngularJS 是 MEAN 堆栈的前端部分，由 MongoDB 数据库，Express.js Web 应用程序服务器框架，Angular.js 本身和 Node.js 服务器运行时环境组成。

AngularJS 的设计目标包括：

将 DOM 操作与应用程序逻辑分离。这样做的难度受到代码结构化的影响。

将应用程序的客户端与服务器端分离。这允许开发工作并行进行，并允许双方重用。

为构建应用程序的旅程提供结构：从设计 UI，通过编写业务逻辑到测试。

Angular 实现了 MVVM 模式来分离表示，数据和逻辑组件。使用依赖注入，Angular 将传统的服务器端服务（如视图相关的控制器）带入客户端 Web 应用程序。因此，可以减少服务器上的大量负担。

在计算机科学中的范围描述了程序中的特定绑定是否有效。ECMA-262 规范将范围定义为：在客户端 Web 脚本中执行 Function 对象的词汇环境；类似于在 lambda 演算中定义范围。

在 AngularJS 中，“范围”是某种类型的对象，其本身可以在程序的任何给定部分中属于范围或超出范围，遵循与其他任何对象相似的 JavaScript 中可变范围的通常规则。当下面使用术语“范围”时，它指的是角度范围对象，而不是名称绑定的范围。

AngularJS 双向数据绑定是其最显着的功能，大大减轻了模板职责的服务器后端。相反，根据模型中定义的范围中包含的数据，模板以纯 HTML 呈现。角度的 \$scope 服务检测模型部分的更改，并通过控制器修改视图中的 HTML 表达式。同样，对该视图的任何改变都反映在模型中。这规避了主动操纵 DOM 并鼓励 Web 应用程序的引导和快速原型化的需要。AngularJS 通过将当前值与脏检查过程中较早存储的值进行比较来检测模型中的更改，与模型值更改时触发侦听器的 Ember.js 和 Backbone.js 不同。

3. MongoDB

MongoDB 支持字段，范围查询，正则表达式搜索。查询可以返回特定字段的文档，还包括用户定义的 JavaScript 函数。查询也可以配置为返回给定大小的结果的随机抽样。

MongoDB 提供了高可用性与副本集。副本集由两个或多个数据副本组成。每个副本集成员可以在任何时候起作用的主副本。默认情况下，所有写入和读取都在主副本上完成。辅助副本使用内置复制维护主数据的副本。当主副本失败时，副本集自动执行选举过程以确定哪个辅助应该成为主要副本。辅助程序可以可选地提供读取操作，但是该数据默认情况下才最终保持一致。

MongoDB 使用分片水平缩放。用户选择一个分片键，确定集合中的数据将如何分发。数据被分割为范围（基于分片键）并分布在多个分片之间。（碎片是一个或多个奴隶的主人）。或者，碎片密钥可以被散列以映射到分片 - 实现均匀的数据分发。

MongoDB 可以在多台服务器上运行，平衡负载或复制数据，以防止硬件故障时系统启动并运行。

MongoDB 可以用作具有负载平衡和多个机器上的数据复制功能的文件系统，用于存储文件。

MongoDB 驱动程序包含了这个名为 Grid File System 的功能。MongoDB 向开发人员公开了文件操作和功能。GridFS 用于 NGINX 和 lighttpd 的插件 GridFS 将文件分成多个部分或块，并将这些块中的每一个作为单独的文档存储。

MapReduce 可用于批量处理数据和聚合操作。

聚合框架使用户能够获得使用 SQL GROUP BY 子句的结果种类。聚合运算符可以一起组成一个管道 - 类似于 Unix 管道。聚合框架包括可以从多个文档中加入文档的 \$lookup 运算符，以及诸如标准偏差的统计运算符。

JavaScript 可用于查询，聚合功能（如 MapReduce），并直接发送到要执行的数据库。

MongoDB 支持固定大小的集合，称为封顶集合。这种类型的集合维护插入顺序，并且一旦达到指定的大小，其行为就像一个循环队列。

4. Webpack

webpack 是用于现代 JavaScript 应用程序的模块绑定器。当 Webpack 处理您的应用程序时，它会递归地构建一个包含应用程序所需的每个模块的依赖关系图，然后将所有这些模块打包成少量浏览器加载的软件包（通常只有一个）。

◆ Entry

webpack 创建了所有应用程序依赖关系的图形。该图的起始点被称为入口点。入口点告诉 webpack 在哪里启动并遵循依赖关系的图表，以了解要捆绑的内容。您可以将应用程序的入口点作为上下文根或第一个文件来启动应用程序。

在 webpack 中，我们使用 webpack 配置对象中的 entry 属性来定义入口点。

◆ Output

将所有资源捆绑在一起后，您仍然需要告知 webpack 在何处捆绑应用程序。webpack 输出属性告诉 webpack 如何处理捆绑代码。

◆ Loaders

目标是使您的项目中的所有资产成为 Webpack 的关注点，而不是浏览器。（这并不意味着他们都必须捆绑在一起）。webpack 将每个文件（.css, .html, .scss, .jpg 等）视为一个模块。但是，webpack 只能理解 JavaScript。

Webpack 中的装载机将这些文件转换为模块，因为它们被添加到依赖关系图中。

在高水平上，它们在您的 webpack 配置中有两个目的。

确定某个装载程序应该转换哪些文件。（测试属性）

转换该文件，以便可以将其添加到依赖关系图（最终将其添加到包中）。（使用属性）

◆ Plugins

由于 Loader 仅在每个文件的基础上执行转换，因此插件最常用于（但不限于）在捆绑模块的“compilations”或“chunks”上执行操作和自定义功能（以及更多）。webpack 插件系统非常强大，可定制。

为了使用插件，您只需要 require（）并将其添加到 plugins 数组。大多数插件可通过选项进行自定义。由于您可以在配置中多次使用插件进行不同的目的，因此您需要通过使用新建来创建一个实例。

第二章 系统需求分析

2.1 系统设计目标

1. 设计一个可供用户了解 Web 开发相关技术，并能够选择相应的技术生成 Web 模版，下载这个 Web 模版的网站。
2. 设计一套 Web 模版，其中包含用户所选的技术文件，以及模块打包管理工具，让用户能够使用这个模版开发 Web 站点。

2.2 系统功能分析

我们的 Web 网站系统为了让用户更好的了解当今 Web 开发的流行技术，并让用户能够任意组合技术生成 Web 框架，从而使用我们的框架实现更快速高效的网站开发。系统功能需求如图所示。

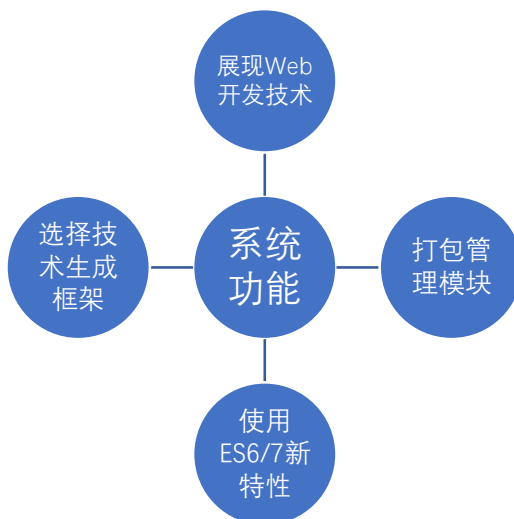


图 2-1 系统功能分析

2.2.1 展现 Web 开发技术功能

用户在使用 web 框架建立网站的时候，首先要对 web 框架里面用到的技术有个深层次的了解，比如前端用到了什么技术，到底适不适合自己的项目需求，在选择 web 框架用到的技术时，要格外严谨，防止出现选择的技术不适合自己的项目需求的情况。

2.2.2 选择技术生成框架功能

用户需要能根据自己对 UI 框架，前端框架和后端框架的选择创建自己的 web 模板。

随着互联网的不断发展，越来越多的 Web 开发技术涌现出来，这个时候我们有必要选择一个好的前端框架与 UI 框架配合，达到减轻开发难度，提高开发效率，美观前端页面，更好管理前端模块的要求。

为了适应网站开发的快速高效的要求，我们后端采用了非阻塞式编程的 Node.js 作为支撑。

当用户将信息填写好，技术选择完毕后，我们后台就会对此进行一系列的处理。第一步：复制文件夹，第二步：根据基本信息和技术的选择改动配置文件，第三步：添加打包管理模块的相关功能，第四步：添加将 ES6/ES7 新语法转换为 ES5 的转换工具，第五步：压缩文件夹，提供用户下载。

2.2.3 打包管理模块功能

用户需要一个好的打包管理工具，对代码进行打包压缩混淆。

随着当今互联网前端技术的疯狂增长，作为开发者使用的前端语言已经不只是 JavaScript, CSS, html 为了适应前端开发的快速高效性，JSX、TypeScript、coffee, less, sass, jade, dust 等技术相应而生，但是我们最终要浏览器运行的还是 JavaScript 语言，这时，我们必须需要一款好的代码打包工具，将这些非 JavaScript 的语言能够打包成为浏览器可运行的 JavaScript 语言。

除此之外，客户在通过浏览器访问网站的时候，需要加载 JavaScript 代码，如果开发的 JavaScript 不进行压缩和混淆，会导致客户访问速度变慢以及安全性的问题，所以一款好的打包工具已经是现在 web 开发的必需品。

2.2.4 使用 ES6/7 新特性功能

用户需要能够使用 ES6/ES7 的新特性，需要能将 ES6/ES7 的语法转换成 ES5 的语法的工具。

ES6/ES7 其实是 JavaScript 的新标准，其中包含了大量的 JavaScript 的新特性，比如变量解构赋值，箭头函数，Promise 等，这些新特性从语法层面上都将使 Web 开发变得更加简单，但是当今浏览器还不能完全兼容这些新的特性，为此我们需要一款能将 ES6/ES7 的语法转化成 ES5 的语法的工具。

2.3 系统可行性分析

可行性分析，是在该系统需求分析完成的基础上，估量系统的开发是否具有可行性，对该系统整个开发从技术、操作、经济的方面进行研究和分析，确认该系统是否可以开发成功以及实施后具备它该有的价值。该系统的可行性分析包括以下几个方面的内容：

1. 操作可行性：

本系统灵活方便，操作简单，界面友好，错误提示明确，便于使用人员快速熟悉和使用。

2. 技术可行性：

该系统对软、硬件系统无特殊要求，软件上，操作系统采用 Windows 系列操作系统，VScode 作为前台开发工具以及 Node.js 作为开发语言，后台使用 MongoDB 数据库，用基于 express 的 kraken.js 框架实现，这些都是现有的主流和成熟的开发工具，语言和框架，而且这些工具和语言都简单易学，开发人员可以较好掌握。

同时，Node.js 编程方便，界面清晰，技术比较成熟。因此在技术上是可行的。

2.4 研究方法与技术路线

研究方法：

首先调查现在互联网 Web 开发的常用技术，分析技术的特点以及学习难度，其次了解 Web 开发的模块打包管理工具，然后进行系统可行性分析与功能分析，然后进行详细设计，编写相关功能模块的源代码，随后进行细致修改，逐步完善系统功能；再进行系统测试，检查系统存在的错误与隐患，并且进行相应的修改，优化网站与 Web 模版，最终完成该系统。

技术路线：

- （1）通过 angularjs 完成前端框架开发

- （2）通过 nodejs 的 express 完成后端框架开发
- （3）通过 Webpack 对各个模块进行管理
- （4）通过 CSS 技术实现模板样式。
- （5）通过 nosql 的 mongodb 对数据进行管理

第三章 系统概要设计

3.1 系统设计思路

现如今，web 开发技术种类繁多，快速高效的开发乃是每个企业追求的目标，很多开发者不知道该如何选择技术，我从 web 开发效率以及开发成本的出发点上面考虑，web 框架后端选择了异步 IO 的 Node.js。Node.js 的主要思路是：使用非阻塞的，事件驱动的 I/O 操作来保持在处理跨平台 (across distributed devices) 数据密集型实时应用时的轻巧高效。相比其他的后端语言 Java，PHP 等，Node.js 显得更加轻量高效，同时学习成本也比较低。框架前端则是采用了现今比较流行的三个前端框架：Angular，Vue，React，用户可以根据自己的需要进行选择。UI 框架这一块 Bootstrap 自然是必不可少的，我们还添加了一款基于 Bootstrap 的 UI 框架 Flat UI，相比 Bootstrap 的 CSS 的样式，它的样式更具扁平化的特点。所以我集合多种 web 开发前沿的技术设计了这一套 web 开发的框架。

3.2 系统功能架构

3.2.1 Web 网站架构

根据需求分析，本系统功能架构大致分为六个部分：关键技术介绍模块，选择技术生成框架模块，Web 框架生成模块，用户下载使用框架模块，系统扩展性设计模块，框架设计模块。用户可以在我们的网站上面第一步了解相关技术，然后选择相应技术生成框架并下载，最后阅读框架的使用文档开始开发。如图所示：

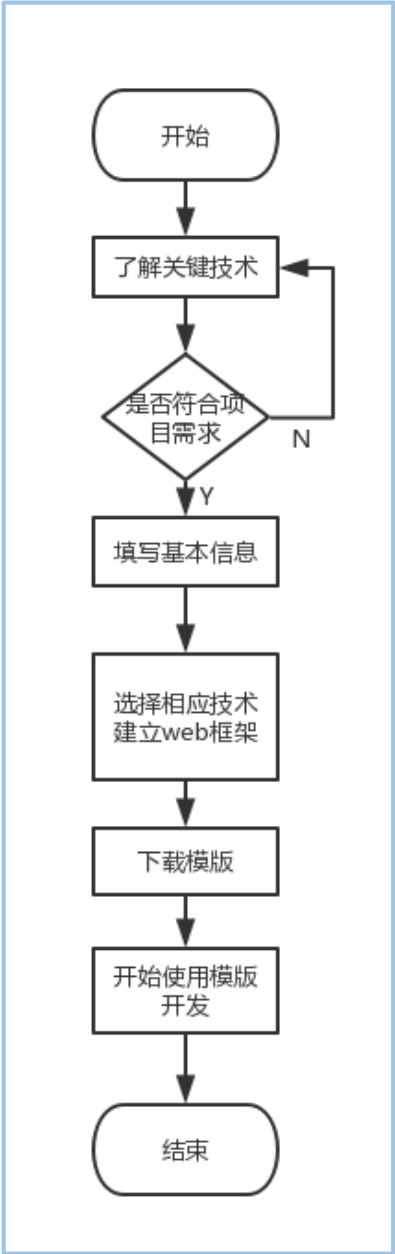


图 3-1 Web 网站功能流程图

3.2.2 Web 框架架构

在设计框架时，主要遵从以下几个原则，如图所示。



图 3-2 框架设计原则

◆ 易用性

在设计框架之初，我们就本着为软件开发企业设计一套使用简单方便的开发框架，通过 Node.js 的简单高效以及前后端统一使用 JavaScript 开发语言，极大提高了开发框架的易用性。

◆ 低成本

Web 自动化开发框架极大的简化了开发流程，不仅提高软件的开发质量，更提高了软件的开发效率，降低了开发成本，这也是我们设计该框架根本的目的。

◆ 技术先进性

在框架设计中，我们采用了 MVC 架构，MVC 框架是用一种业务逻辑、数据、界面显示分离的软件架构，将业务逻辑聚集到一个部件里面，在改进和个性化定制界面及用户交互的同时，不需要重新编写业务逻辑。其次，我们都采用了现在 Web 开发的主流技术 AngularJs, Vue, React, Node.js, Bootstrap 等开发了一套轻量级 Web 开发框架。应用该开发框架可以有效提升软件开发效率，降低开发成本。

◆ 可扩展性

利用我们的开发框架，开发人员可以根据自己的项目需求修改配置文件添加相应 Web 模块。我们也会将市场上一些新的技术加入到我们的网站以及开发框架中。

在框架的开发中，我们的后端主要围绕控制器、视图、模型进行组件的划分，前端主要是 MVVM 架构的体现，我们的 Web 框架大体的 MVC 架构如图所示。

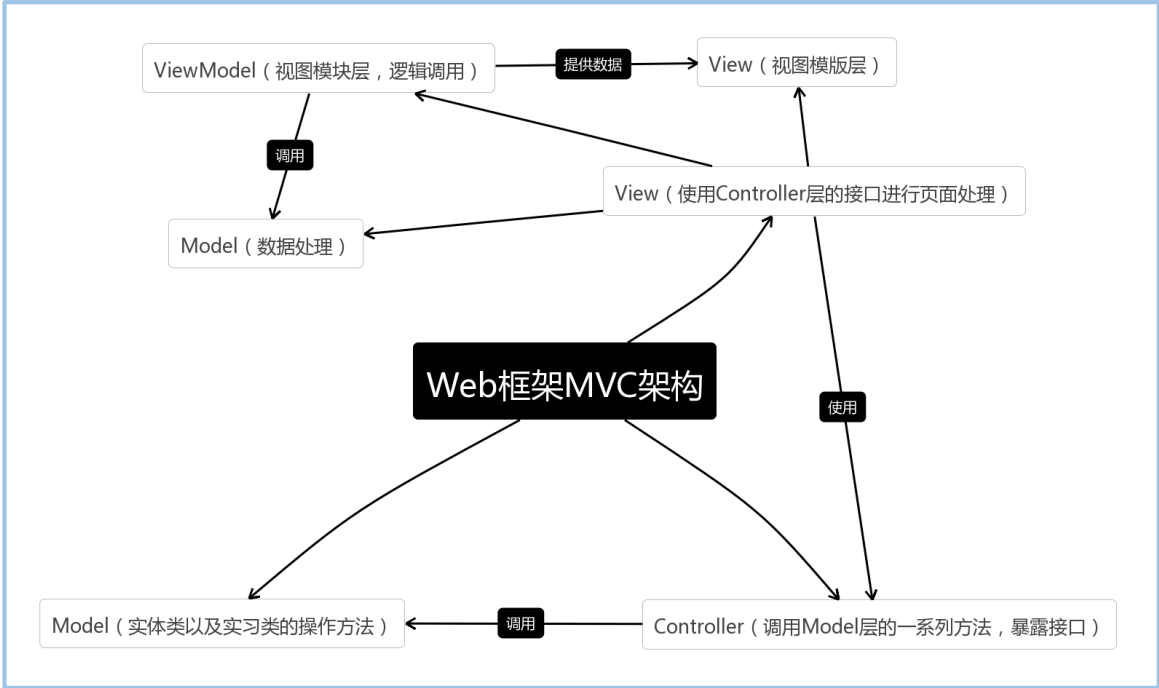


图 3-3 Web 框架 MVC 架构图

后端 MVC 的全名是 Model-View-Controller。MVC 是用于在计算机上的一种软件架构模式。它将一个给定的应用程序划分为三个相互连接的部分，以便将信息的内部表示与信息的呈现方式分离并从用户接受。MVC 设计模式使这些主要组件分离，从而实现高效的代码重用和并行开发。

传统上用于桌面图形用户界面（GUI），这种架构已经变得流行于设计 Web 应用程序，甚至移动，桌面和其他客户端。流行的编程语言（如 Java，C#，Ruby，PHP 等）具有流行的 MVC 框架，目前正在 Web 应用程序开发中直接使用。下面是 MVC 设计模式的关系图。

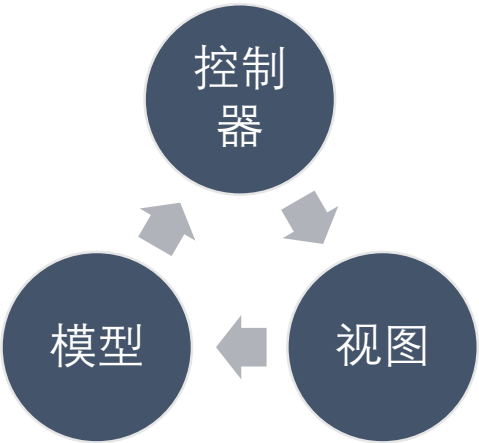


图 3-4 MVC 关系图

◆ 模型

模型是模式的核心组成部分。它表示应用程序在问题域方面的行为，与用户界面无关。它直接管理应用程序的数据，逻辑和规则。

◆ 视图

视图可以是信息的任何输出表示，例如图表或图表。相同信息的多个视图是可能的，例如用于管理的条形图和会计师的表格视图。

◆ 控制器

控制器接受输入并将其转换为模型或视图的命令。

前端 MVVM 的全名为 Model-view-viewmodel，它是一种软件架构模式。MVVM 有助于通过开发业务逻辑或后端逻辑（数据模型），分离图形用户界面的开发（无论是通过标记语言还是 GUI 代码）。MVVM 的视图模型是一个值转换器；意味着视图模型负责从模型中显示（转换）数据对象，使得对象容易管理和呈现。在这方面，视图模型比视图更模型，并处理大部分（如果不是全部）视图的显示逻辑。视图模型可以实现调解器模式，组织对视图支持的一组用例的后端逻辑的访问。MVVM 是 Martin Fowler 的演示模型设计模式的一个变体。MVVM 以同样的方式提取视图的状态和行为，但是 Presentation Model 会以不依赖于特定用户界面平台的方式抽象视图（创建视图模型）。MVVM 和 Presentation Model 都来源于模型 - 视图 - 控制器模式（MVC）

◆ 模型

模型是指代表真实状态内容（面向对象的方法）的域模型，也可以是代表内容（以数据为中心的方法）的数据访问层。

◆ 视图

与 MVC 和 MVP 模式一样，视图是用户在屏幕上看到的结构，布局和外观。（UI）

◆ 视图模型

视图模型是暴露公共属性和命令的视图的抽象。代替 MVC 模式的控制器或 MVP 模式的演示者，MVVM 具有绑定器。在视图模型中，粘合剂介导视图和数据绑定器之间的通信。视图模型已被描述为模型中数据的状态。

◆ 绑定器

声明式数据和命令绑定在 MVVM 模式中是隐含的。在 Microsoft 解决方案堆栈中，**binder** 是一种名为 XAML 的标记语言。绑定器使开发人员不必编写锅炉板逻辑来同步视图模型和视图。当在 Microsoft 堆栈之外实现时，声明性数据绑定技术的存在是该模式的关键推动因素。

第四章 系统详细设计

详细设计阶段的根本目标是确定应该如何具体地实现所要求的系统，经过该阶段的设计工作，应该得出对目标系统的精确描述，详细设计的结果决定了这个项目最终的质量。下面根据概要设计的五大功能模块依次描述。

4.1 关键技术介绍模块

首先要对当今互联网上面的 Web 开发技术做一个采集，然后根据使用的流程度，技术的特点以及学习的难度筛选出我们要使用的技术。再将这些技术的存放的我们的 Mongoddb 数据库中，前端在通过 AJAX 从后端暴露的 web 开发技术的接口中拿到数据，前端在通过 AngularJS 和 Bootstrap 框架将这些技术展现出来，这样用户就可以根据自己的需要了解并学习这些技术，为以后的选择技术搭建框架做准备。

本地 Mongoddb 数据库数据表如下：

Web 开发技术表：

字段	数据类型	含义
_id	ObjectId	技术 ID
Name	String	技术名称
Detail	String	技术详情
Href	String	技术官网链接
Kind_id	number	技术类别

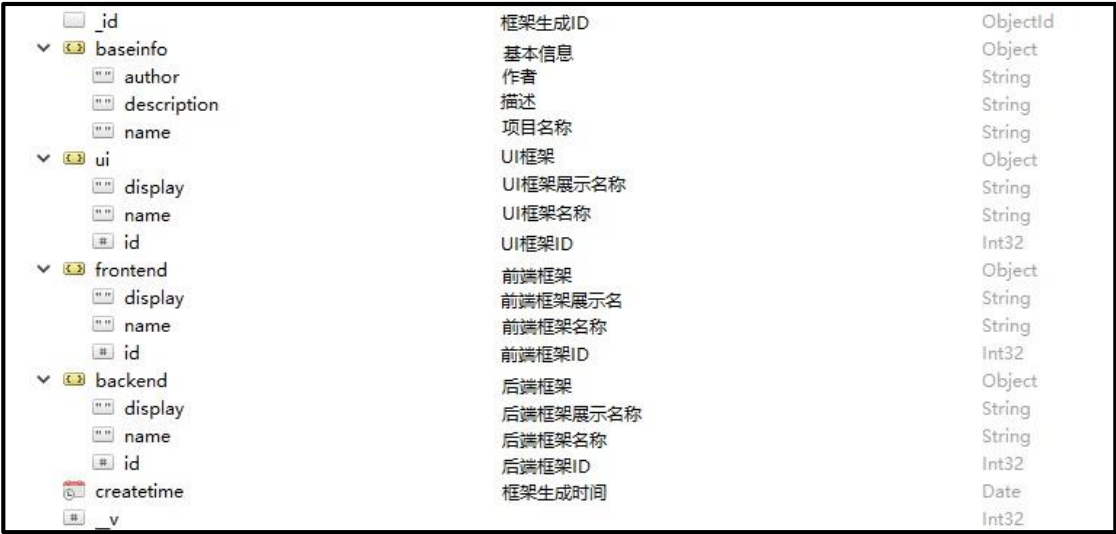
表 4-1 Web 开发技术表

4.2 选择技术生成框架模块

在这个模块上面，用户需要根据自己的需求，填写相应的信息，并通过按钮点击选择相应的技术，然后通过 AJAX 的 POST 请求发送到数据库，这样我们的框架生成表就能够添加一条记录。

本地 Mongoddb 数据库数据表如下：

框架生成表：



_id	框架生成ID	ObjectId
baseinfo	基本信息	Object
author	作者	String
description	描述	String
name	项目名称	String
ui	UI框架	Object
display	UI框架展示名称	String
name	UI框架名称	String
id	UI框架ID	Int32
frontend	前端框架	Object
display	前端框架展示名	String
name	前端框架名称	String
id	前端框架ID	Int32
backend	后端框架	Object
display	后端框架展示名称	String
name	后端框架名称	String
id	后端框架ID	Int32
createtime	框架生成时间	Date
v		Int32

图 4-2 框架生成表

4.3 Web 框架生成模块

这一个模块就是最重要的一部分，就是根据用户的信息以及选择的技术生成相应的 Web 框架。

1. 根据用户选择的后端框架，调用相应的后端的方法，然后生成相应的文件夹。
2. 根据用户填写的信息以及选择的技术，在生成出来的文件夹里面修改配置文件（package.json），添加用户选择的相应技术模块。
3. 为了让用户更好的对代码进行打包和管理，我们在配置文件（package.json）中添加了 Webpack 模块，以及在项目中添加 Webpack 的配置文件 webpack.config.js，它不仅能对 JavaScript 代码进行压缩混淆，还能把类 JavaScript 语言代码转换成 JavaScript 代码，以及将 Less/Sass 转换成 CSS 文件。
4. 为了能让用户使用 ES6/ES7 的新特性，我们在配置文件（package.json）中添加了 babel 的相应模块，以及添加 babel 配置文件.babelrc，将 ES6/ES7 的代码转换成浏览器可以运行的 ES5 的代码。
5. 最后我们将生成的文件夹打成 Zip 压缩包，提供给用户进行下载使用。

流程图如下：

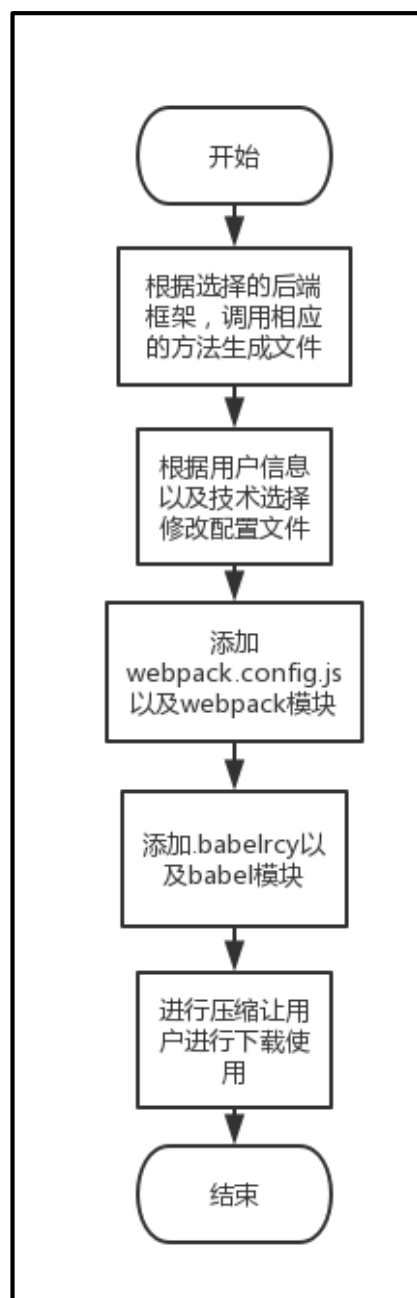


图 4-3 生成框架流程图

4.4 用户下载使用框架模块

用户通过我们的 Web 站点下载好创建的 Web 的框架之后，首先安装好 Node.js 环境，然后在通过 NPM 安装各个模块，然后仔细阅读其中的文档，了解各个目录的意义以及在项目中的作用，这样才能够更快速的使用这套 Web 框架开发站点。

4.5 系统扩展性设计模块

我们的这套 Web 框架不局限于网站上陈列的技术，如果以后会有另外的技术添加进来，我们只需在数据库中添加这些技术，然后在 resource 文件夹中生成好目录结构，如果是后端框架，则会根据这个后端框架再生成一套相应的目录结构，如果是前端框架或者 UI 框架，只需要修改项目配置文件（package.json）。用户也可以自定义添加技术，只需根据上面的步骤进行操作。

4.6 框架设计模块

首先我们的 Web 框架完全是前后端分离的，易于前后端各个模块的分离，满足了软件开发中各个模块的高内聚低耦合的要求，这样各个模块才能易于管理与代码的重用，也能够更好的实现多人合作开发的需求，完全满足了当今 Web 网站开发的潮流与趋势。

4.6.1 框架前端设计

Web 框架前端采用的 MVVM 架构，框架图如下所示：

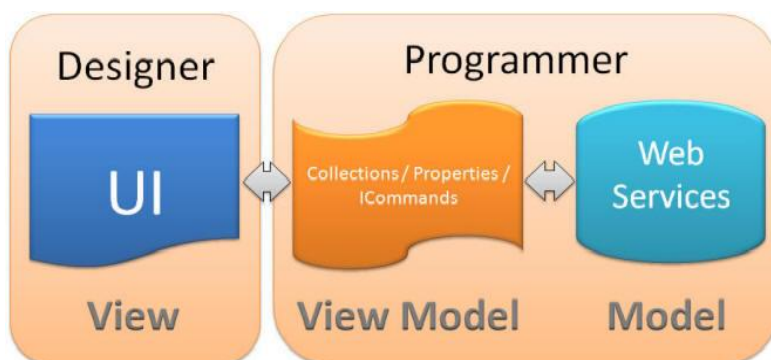


图 4-6-1 MVVM 框架

- ◆ Model 层主要为应用程序提供数据，它能够通过 AJAX 向后端接口发出请求从而获取数据或者其他一系列操作
- ◆ View Model 主要的就是负责联通 View 层和 Model 层，从 Model 层拿到的数据需要做一些业务逻辑的处理才能够显示到 View 层上面。
- ◆ View 层主要是负责页面的排版和布局，将 View Model 层处理完的数据展现到前台页面上。

4.6.2 框架后端设计

Web 框架后端采用 MVC 架构，具体目录结构如下图所示

configs	2017/5/23 22:02	文件夹	
controllers	2017/5/23 22:02	文件夹	
kits	2017/5/23 22:02	文件夹	
middlewares	2017/5/23 22:02	文件夹	
models	2017/5/23 22:02	文件夹	
public	2017/5/23 22:02	文件夹	
routers	2017/5/23 22:02	文件夹	
test	2017/5/23 22:02	文件夹	
views	2017/5/23 22:02	文件夹	
.babelrc	2017/5/30 0:33	BABELRC 文件	0 KB
.eslintrc.js	2016/9/5 16:34	JavaScript 文件	1 KB
.gitignore	2016/9/5 16:34	GITIGNORE 文件	1 KB
.jsbeautifyrc	2016/9/5 16:34	JSBEAUTIFYRC ...	1 KB
app.js	2016/9/5 16:34	JavaScript 文件	1 KB
gulpfile.js	2016/9/5 16:34	JavaScript 文件	5 KB
package.json	2016/9/5 16:34	JSON 文件	2 KB
README.md	2016/9/5 16:34	MD 文件	2 KB
webpack.config.js	2017/5/24 11:25	JavaScript 文件	2 KB

图 4-6-2 框架后端目录图

从上至下，介绍该框架结构：

1. Configs 这里主要是框架后端的一些配置，比如数据库的配置，中间件的配置，开发模式的配置等等。
2. Controllers 这里主要是暴露给前端的一些接口，可供前端通过 http 请求访问到数据。
3. Kits 这里是一些集成的软件包
4. Middlewares 这里是 Node.js 的特有的中间件配置，里面可以存放 Node.js 的一些中间件，比如像用户权限认证的一些中间件等。
5. Models 这里存放的一些数据模型和数据的一些操作方法。
6. Public 这里存放一些项目的静态文件，一些图片，CSS，JavaScript 代码都可以放在这个目录下。
7. Routers 这里主要是暴露给前端的一些路由接口，可供前端通过 http 请求访问。
8. Test 这是集成测试和单元测试的目录。

9. Views 这里是一些前端的模版文件，用来展示前端页面，比如 dust, jade 等。
10. .babelrc 这里就是 babel 的配置文件，里面可以将指定的 ES6/7 的代码转换成 ES5 的代码，这样浏览器可以兼容执行。
11. README.md 这里介绍了系统目录的架构，开发者需阅读来使用该 Web 框架。
12. Webpack.config.js 这里是 webpack 的配置文件，可以将配置好的文件代码进行压缩混淆，也可以加载一些解析器，完成一些特定的功能，比如将 less/sass 转换成 CSS。

第五章 系统实现

5.1 系统展示

1. 技术介绍页面

技术介绍页面是本网站的主页，这个页面展示了一些当今 Web 开发流行的技术，包括这些技术的简要介绍，特点，以及技术的官方网址，这样用户就可以根据自己的需要了解并学习这些技术，为以后的选择技术搭建框架做准备。技术介绍页面如下图所示。



图 5-1 技术介绍页面

2. 创建框架页面

用户可以通过这个页面，填写项目的基本信息，然后选择自己要用到的各种技术，生成并下载框架。如下图所示。

Vanguard

主页

创建框架

Webpack介绍

基础信息

Project Name

Project Name

Description

Description

Author

Author

UI框架

Bootstrap

Flat UI

前端框架

AngularJs

Vue

React

后端框架

Express

Koa.js

Submit and Download

图 5-2 创建框架页面

3. Webpack 介绍页面

因为 Webpack 是我们项目的一种重要组成部分，所以我们用了单独的一个页面来重点介绍 Webpack，这样用户才能了解到 Webpack 的功能，更好的使用 Webpack 进行项目的模块化管理以及打包压缩等功能。Webpack 介绍页面如下图所示。



图 5-3 Webpack 介绍页面

5.2 系统编码

1. 前端开发用 AngularJs 和 Bootstrap 结合，具体代码如下：

a) 路由模块：

```
$stateProvider
    .state('index', {
        url: "/index",
        views:{
            'title':{
                templateUrl: '/app/carousel.html',
            },
            'content':{
                templateUrl: '/app/index.html',
            }
        }
    })
    .state('order', {
        url: "/order",
        views:{
```

```

        'content': {
            templateUrl: '/app/order.html',
        }
    }
})

```

b) 前端页面:

```

<h2 class="row">UI 框架</h2>
<div class="row col-xs-offset-2">
    <div class="form-group">
        <div class="col-xs-4"
            ng-repeat="ui in uis">
            <a class="btn btn-lg btn-outline"
                ng-class="{ 'selected' :
                    order.ui==ui}"
                ng-click="order.ui=ui">
                {{ui.display}}
            </a>
        </div>
    </div>
</div>
<hr/>
<h2 class="row">前端框架</h2>
<div class="row col-xs-offset-2">
    <div class="form-group">
        <div class="col-xs-4"
            ng-repeat="frontend in frontends">
            <a class="btn btn-lg btn-outline"
                ng-class="{ 'selected' :
                    order.frontend==frontend}"
                ng-click="order.frontend=frontend">
                {{frontend.display}}
            </a>
        </div>
    </div>
</div>
<hr/>

```

```

<h2 class="row">后端框架</h2>
<div class="row col-xs-offset-2">
  <div class="form-group">
    <div class="col-xs-4"
      ng-repeat="backend in backends">
      <a class="btn btn-lg btn-outline"
        ng-class="' selected' :
          order.backend==backend}"
        ng-click="order.backend=backend">
        {{backend.display}}
      </a>
    </div>
  </div>
</div>

```

c) 前端样式:

```

.btn-outline {
  color: $color1;
  background-color: transparent;
  border-color: $color1;
}
.btn-outline:active, .btn-outline:focus, .btn-outline:hover {
  color: #fff;
  background-color: $color1;
  border-color: $color1;
}
.selected{
  color: #fff;
  background-color: $color1;
  border-color: $color1;
}
.form-control{
  color: $color1 !important;
  border: 1px solid $color1;
}
.form-control:focus{
  border-color: $color2;
}

```

```
}  
.carousel-caption{  
padding-bottom: 100px;  
}
```

2. 后端用 Node.js 开发, 建立 restful Api 供前端 ajax 访问, 数据库为 mongodb:

a) 接口路由:

```
router.get('/', function(req, res) {  
    book.find(function(err, doc) {  
        res.json(doc);  
    })  
});  
router.post('/add', function(req, res) {  
    var newBook=new book(req.body.order);  
    newBook.save(function(err, doc) {  
        if(err) {  
            res.json(err);  
        }else{  
            res.json(newBook);  
        }  
    })  
});
```

b) 底层数据库操作:

```
var mongoose = require('mongoose');  
var book = require('../models/module').order;  
mongoose.connect('mongodb://localhost/hello');  
var newBook=new book(req.body.order);  
    newBook.save(function(err, doc) {  
        if(err) {  
            res.json(err);  
        }else{  
        }  
    })
```

c) 文件操作:

```
express: function(order, callback) {  
    var time = Date.parse(new Date());  
    flieService().copyfile(path + order.backend.name,
```

```
path + order.backend.name + time, function () {
    var packagejson = require(path +
    order.backend.name + time + "\\package.json");
    packagejson.name = order.baseinfo.name;
    packagejson.description = order.baseinfo.description;
    packagejson.author = order.baseinfo.author;
    packagejson.dependencies = "";
    packagejson.dependencies = "";
    fs.writeFile(path + order.backend.name +
    time + "\\package.json",
    JSON.stringify(packagejson).replace(/,/g, ",\n"),
    function (err) {
        if (err) {
            throw err;
        } else {
            flieService().zip(order.backend.name + time,
            path + order.backend.name + time, function () {
                callback(order.backend.name + time);
            })
        }
    });
})
},
```

第六章 结论

全文总结

通过一段时间的努力，我顺利完成了《基于 Node.js 的 Web 框架生成工具》毕业设计。在这次毕业论文设计过程中我通过上网查询，翻阅书籍，向老师和同学请教相关知识等方法解决了很多遇到的问题，学习到了很多知识，同时也认识到自己在技术方面有很多的不足，认识到理论结合实践的重要性，认识到在开发网站时应事先计划好开发的每一步骤。

本系统按照需求分析的功能基本实现，但是还有些功能需要优化，通过这段时间的毕业设计的实现及别与论文的撰写，我深知自己还有许多的方面需要学习和改正，在以后的学习中，会更加完善自己。

展望

本系统是基于 Node.js 的 Web 框架生成工具，通过展现 Web 开发的常用技术以及根据用户的选择生成相应的 Web 框架，开发人员能够使用框架进行快速高效的网站开发，现对未来可以完善系统的工作做出几点展望：

1. 本系统收集的暂时还是 Web 开发较为流行的一些技术，有些新技术还未收集齐全。
2. 希望能够在未来的学习中，增强自己的能力，对此系统进行升级或改造，使得生成的 Web 框架更加强大，能够实现一些 Web 开发的常用功能。

相信随着研究的深入，这些问题将会得到很好的解决，相信基于 Node.js 的 Web 框架生成工具这套系统会有很好的发展前景。

参考文献

- [1] 黄扬子 基于 NodeJS 平台搭建 REST 风格 Web 服务 无线互联科技 2015-08-25
- [2] 许会元; 何利力 NodeJS 的异步非阻塞 I/O 研究 工业控制计算机 2015-03-25
- [3] 柯肇丰; 曾霞霞 基于 HTML5+nodeJS+MongoDB 构建在线图像编辑器系统 福建电脑 2015-06-25
- [4] 郑立国; 于宏伟 基于 Nodejs 的本机应用与 Web 应用统一编程模型的研究 优先出版电子技术与软件工程 2015-06-16
- [5] 郑杰 AngularJS 前端框架与后端代码自动生成系统的设计与实现 西南交通大学 2017-05-10
- [6] 冯晨超 基于 AngularJS 的物品和用户查询模块的设计与实现 南京大学 2015-05-01
- [7] 张文盛; 郑汉华 基于 MongoDB 构建高性能网站技术研究 吉林师范大学学报(自然科学版) 2013-02-10
- [8] 黄悦深 基于 HTML5 的移动 Web App 开发 图书馆杂志 2014-07-15
- [9] 杨俊; 李艳梅 JavaScript 面向对象编程探析 办公自动化 2010-04-15
- [10] 张俊兰; 刘翼; 铁宏军 基于 CSS 技术的网页设计应用研究 延安大学学报(自然科学版) 2010-09-20
- [11] 周玲余 基于 jQuery 框架的页面前端特效的设计与实现 计算机与现代化 2013-01-15
- [12] Seonghoon Kang; Sukyoung Ryu ACM SIGPLAN Notices, 2012, Vol.47 (10), pp.621-638
Association for Computing Machinery Journal
- [13] Lubbers P, Albers B, Salim F Pro HTML5 Programming: Powerful APIs for Richer Internet
Application Development 2010
- [14] Tilkov, Stefan, Vinoski, Steve. Node.js: Using JavaScript to Build High-Performance Network
Programs[J]. IEEE Internet Computing . 2010 (6)
- [15] Erasmus M. .CoffeeScript Programming with JQuery, Rails, and Node. js. 2012
- [16] Rauch G. Smashing Node. js: JavaScript Everywhere. 2012

致谢

在这一段时间内，我能成功的将此毕业设计做出，这要归功于朱晓辉老师严格的要求、合理的时间安排以及程序的技术支持。老师在这期间对我的帮助和指导，更令我受益匪浅，在此表示深深地感谢。本文是在朱晓辉老师的关心和指导下完成的，在论文完成过程中，本人还得到了其他同学与朋友的热心帮助，本人向他们表示深深的谢意！