# 基于nodejs的台账管理系统

# 绪论

### 1.1研究背景和意义

##### 1.1.1研究背景

当今社会对统计工作的需求越来越大，对统计工作的要求也越来越高。面对挑战，

如何进一步提高基层统计工作的规范性和工作效率，一直是各级统计部门领导关注的焦点。统计工作的准确性需要从数据源头抓起，而基础统计台账的登记正是做实数据源头的有效抓手。如何建立一套高效实用的统计电子台账，并及时有效的推广和引导广大企事业单位建立健全统计台账制度是一个值得研究的课题，也是本文所讨论的内容。

从行业发展来看，我国统计部门信息管理已经有了长足的发展，但与国际竞争还存在一定差距，主要面临以下问题：从技术与管理层面来看，我国统计数据管理，尤其是台账的管理手动操作成分较多，信息化程度较低。

随着信息技术的不断发展，统计工作也逐渐在从手工填报、纸质化向自动填报、网络化转变。2002年，杭州市就已经开发和推广了“企业网上直报系统”和“统计综合数据库系统”，使用以来得到了各级统计部门和直报企业的认可；2012年，全国企业直报系统和数据处理系统“国家一套表系统”在全国范围内推广使用开，要求全国规模以上企业全部实现网上直报。

直报企业上报数据需要从各部门台账中提取整理数据，登记繁琐、保存不便的传统纸质台账己不能满足基层统计的工作需要。由于大部分企业填报统计报表的原始数据都来源于车间生产订单和财务系统，最原始的电子统计台账可以认为是企业统计人员个人整理的电子表格系统或者是企业ERP采购系统和财务软件等早期企业一开始使用的软件。但这类表格和软件并不能十分直接和系统的满足报表上报功能，开发和推广统计电子台账，是统计部门进一步规范基础建设，确保统计质量，提高统计信息化综合水平的有效尝试。

信息技术在统计工作中己经得到了广泛的应用，信息化的统计基础台帐取代手工台账己成为统计基础工作发展的必然趋势。《基于nodejs台账管理系统》的推广使用，体现了统计工作中努力实现高效工作的目标要求，符合全国和全省统计改革发展的方向和思路。实现统计台账电子化，是服从服务于科技创新发展的迫切需要，统计工作适应要求、服务发展的重要体现，是推进统计制度方法改革创新的重要举措，是提高统计数据质量的技术保障，是加强统计基层基础建设的有效抓手。

##### 1.1.2研究意义

### 1.2研究现状

### 1.3研究内容

### 1.4本章小结

# 相关技术概述

### 2.1 B/S模式

浏览器-服务器（Browser/Server）结构，简称[B/S结构](https://zh.wikipedia.org/wiki/B/S%E7%BB%93%E6%9E%84" \o "B/S结构)，与[C/S结构](https://zh.wikipedia.org/wiki/C/S%E7%BB%93%E6%9E%84" \o "C/S结构)不同，其客户端不需要安装专门的[软件](https://zh.wikipedia.org/wiki/%E8%BD%AF%E4%BB%B6" \o "软件)，只需要[浏览器](https://zh.wikipedia.org/wiki/%E6%B5%8F%E8%A7%88%E5%99%A8" \o "浏览器)即可，浏览器与[Web](https://zh.wikipedia.org/wiki/Web" \o "Web)[服务器](https://zh.wikipedia.org/wiki/%E6%9C%8D%E5%8A%A1%E5%99%A8" \o "服务器)交互，[Web](https://zh.wikipedia.org/wiki/Web" \o "Web)[服务器](https://zh.wikipedia.org/wiki/%E4%BC%BA%E6%9C%8D%E5%99%A8" \o "服务器)与后端[数据库](https://zh.wikipedia.org/wiki/%E6%95%B0%E6%8D%AE%E5%BA%93" \o "数据库)进行交互，可以方便地在不同平台下工作；服务器端可采用高性能[计算机](https://zh.wikipedia.org/wiki/%E8%AE%A1%E7%AE%97%E6%9C%BA" \o "计算机)，并安装[Oracle Database](https://zh.wikipedia.org/wiki/Oracle_Database" \o "Oracle Database)、[DB2](https://zh.wikipedia.org/wiki/DB2" \o "DB2)、[MySQL](https://zh.wikipedia.org/wiki/MySQL" \o "MySQL)等数据库。B/S结构简化了客户端的工作，它是随着[Internet](https://zh.wikipedia.org/wiki/Internet" \o "Internet)技术兴起而产生的，对C/S技术的改进，但该结构下服务器端的工作较重，对服务器的性能要求更高。

B/S架构采取浏览器请求，服务器响应的工作模式。用户可以通过浏览器去访问Internet上由Web服务器产生的文本、数据、图片、动画、视频点播和声音等信息；而每一个Web服务器又可以通过各种方式与数据库服务器连接，大量的数据实际存放在数据库服务器中；从[Web服务](https://baike.baidu.com/item/Web%E6%9C%8D%E5%8A%A1/2837593" \t "https://baike.baidu.com/item/B/S%E7%BB%93%E6%9E%84/_blank)器上下载程序到本地来执行，在下载过程中若遇到与数据库有关的指令，由Web服务器交给数据库服务器来解释执行，并返回给Web服务器，Web服务器又返回给用户。在这种结构中，将许许多多的网连接到一块，形成一个巨大的网，即全球网。而各个企业可以在此结构的基础上建立自己的Internet。

在 B/S 模式中，用户是通过浏览器针对许多分布于网络上的服务器进行请求访问的，浏览器的请求通过服务器进行处理，并将处理结果以及相应的信息返回给浏览器，其他的数据加工、请求全部都是由Web Server完成的。通过该框架结构以及植入于操作系统内部的浏览器，该结构已经成为了当今软件应用的主流结构模式。

B/S架构最大的优点是总体拥有成本低、维护方便、 分布性强、开发简单，可以不用安装任何专门的软件就能 实现在任何地方进行操作，客户端零维护，系统的扩展非常容易，只要有一台能上网的电脑就能使用。

B/S架构是从C/S架构改进而来，可以说是三层C/S架构，由此可见两者关系不一般。B/S从C/S中脱离而出，后来随着WEB技术的飞速发展以及人们对网络的依赖程度加深，B/S一举成为当今最流行的网络架构。两种架构都在各自岗位上虎虎生威，它们各有千秋，都是非常重要的网络架构。在响应速度，用户界面，数据安全等方面，C/S强于B/S，但是在业务扩展和适用www条件下，B/S明显胜过C/S。可以这么说，B/S的强项就是C/S的弱项，反之亦然。它们各有优缺点，相互无法取代。

### 2.2 JavaScript语言概述

JavaScript（通常缩写为JS）是一种[高级](https://zh.wikipedia.org/wiki/%E9%AB%98%E7%BA%A7%E8%AF%AD%E8%A8%80" \o "高级语言)的、[解释型](https://zh.wikipedia.org/wiki/%E7%9B%B4%E8%AD%AF%E8%AA%9E%E8%A8%80" \o "解释型语言)的[编程语言](https://zh.wikipedia.org/wiki/%E7%BC%96%E7%A8%8B%E8%AF%AD%E8%A8%80" \o "编程语言)。JavaScript是一门[基于原型](https://zh.wikipedia.org/wiki/%E5%9F%BA%E4%BA%8E%E5%8E%9F%E5%9E%8B%E7%BC%96%E7%A8%8B" \o "基于原型编程)、[头等函数](https://zh.wikipedia.org/wiki/%E5%A4%B4%E7%AD%89%E5%87%BD%E6%95%B0" \o "头等函数)的语言，是一门多范式的语言，它支持[面向对象](https://zh.wikipedia.org/wiki/%E9%9D%A2%E5%90%91%E5%AF%B9%E8%B1%A1%E7%A8%8B%E5%BA%8F%E8%AE%BE%E8%AE%A1" \o "面向对象程序设计)程序设计，[指令式编程](https://zh.wikipedia.org/wiki/%E6%8C%87%E4%BB%A4%E5%BC%8F%E7%BC%96%E7%A8%8B%E8%AF%AD%E8%A8%80" \o "指令式编程语言)，以及[函数式编程](https://zh.wikipedia.org/wiki/%E5%87%BD%E6%95%B0%E5%BC%8F%E7%BC%96%E7%A8%8B%E8%AF%AD%E8%A8%80" \o "函数式编程语言)。它提供语法来操控文本、[数组](https://zh.wikipedia.org/wiki/%E6%95%B0%E7%BB%84" \o "数组)、日期以及[正则表达式](https://zh.wikipedia.org/wiki/%E6%AD%A3%E5%88%99%E8%A1%A8%E8%BE%BE%E5%BC%8F" \o "正则表达式)等，不支持[I/O](https://zh.wikipedia.org/wiki/I/O" \o "I/O)，比如网络、存储和图形等，但这些都可以由它的宿主环境提供支持。它已经由[ECMA（欧洲电脑制造商协会）](https://zh.wikipedia.org/wiki/Ecma%E5%9B%BD%E9%99%85" \o "Ecma国际)通过[ECMAScript](https://zh.wikipedia.org/wiki/ECMAScript" \o "ECMAScript)实现语言的标准化。它被世界上的绝大多数网站所使用，也被世界主流[浏览器](https://zh.wikipedia.org/wiki/%E6%B5%8F%E8%A7%88%E5%99%A8" \o "浏览器)（[Chrome](https://zh.wikipedia.org/wiki/Google_Chrome" \o "Google Chrome)、[IE](https://zh.wikipedia.org/wiki/Internet_Explorer" \o "Internet Explorer)、[Firefox](https://zh.wikipedia.org/wiki/Firefox" \o "Firefox)、[Safari](https://zh.wikipedia.org/wiki/Safari" \o "Safari)、[Opera](https://zh.wikipedia.org/wiki/Opera%E9%9B%BB%E8%85%A6%E7%80%8F%E8%A6%BD%E5%99%A8" \o "Opera电脑浏览器)）支持。

JavaScript与[Java](https://zh.wikipedia.org/wiki/Java" \o "Java)在名字或语法上都有很多相似性，但这两门编程语言从设计之初就有很大的不同，JavaScript的语言设计主要受到了[Self](https://zh.wikipedia.org/wiki/Self" \o "Self)（一种基于原型的编程语言）和[Scheme](https://zh.wikipedia.org/wiki/Scheme" \o "Scheme)（一门函数式编程语言）的影响。在语法结构上它又与[C语言](https://zh.wikipedia.org/wiki/C%E8%AF%AD%E8%A8%80" \o "C语言)有很多相似（例如if条件语句、switch语句、while循环、do-while循环等）。

在客户端，JavaScript在传统意义上被实现为一种解释语言，但在最近，它已经可以被[即时编译](https://zh.wikipedia.org/wiki/%E5%8D%B3%E6%99%82%E7%B7%A8%E8%AD%AF" \o "即时编译)（JIT）执行。随着最新的[HTML5](https://zh.wikipedia.org/wiki/HTML5" \o "HTML5)和[CSS3](https://zh.wikipedia.org/wiki/CSS3" \o "CSS3)语言标准的推行它还可用于游戏、桌面和移动应用程序的开发和在服务器端网络环境运行 如[Node.js](https://zh.wikipedia.org/wiki/Node.js" \o "Node.js)。

### 2.3 Node.js环境概述

Node.js 是能够在[服务器](https://zh.wikipedia.org/wiki/%E4%BC%BA%E6%9C%8D%E5%99%A8" \o "服务器)端运行 [JavaScript](https://zh.wikipedia.org/wiki/JavaScript" \o "JavaScript) 的[开放源代码](https://zh.wikipedia.org/wiki/%E9%96%8B%E6%94%BE%E5%8E%9F%E5%A7%8B%E7%A2%BC" \o "开放源代码)、[跨平台](https://zh.wikipedia.org/wiki/%E8%B7%A8%E5%B9%B3%E5%8F%B0" \o "跨平台)[执行环境](https://zh.wikipedia.org/wiki/%E6%89%A7%E8%A1%8C%E7%8E%AF%E5%A2%83" \o "执行环境)。Node.js 由 OpenJS Foundation（原为 Node.js Foundation，已与 JS Foundation 合并）持有和维护，亦为 [Linux 基金会](https://zh.wikipedia.org/wiki/Linux_%E5%9F%BA%E9%87%91%E4%BC%9A" \o "Linux 基金会)的项目。Node.js 采用 [Google](https://zh.wikipedia.org/wiki/Google" \o "Google) 开发的 [V8](https://zh.wikipedia.org/wiki/V8_(JavaScript%E5%BC%95%E6%93%8E)" \o "V8 (JavaScript引擎)) 执行代码，使用[事件驱动](https://zh.wikipedia.org/wiki/%E4%BA%8B%E4%BB%B6%E9%A9%85%E5%8B%95" \o "事件驱动)、[非阻塞](https://zh.wikipedia.org/w/index.php?title=%E9%9D%9E%E9%98%BB%E5%A1%9E&action=edit&redlink=1" \o "非阻塞（页面不存在）)和[异步输入输出](https://zh.wikipedia.org/wiki/%E5%BC%82%E6%AD%A5IO" \o "异步IO)模型等技术来提高性能，可优化应用程序的传输量和规模。这些技术通常用于资料密集的即时应用程序。

Node.js 大部分基本模块都用 JavaScript 语言编写。在 Node.js 出现之前，JavaScript 通常作为客户端程序设计语言使用，以JavaScript 写出的程序常在用户的浏览器上执行。Node.js 的出现使 JavaScript 也能用于服务端编程。Node.js 含有一系列内置模块，使得程序可以脱离 [Apache HTTP Server](https://zh.wikipedia.org/wiki/Apache_HTTP_Server" \o "Apache HTTP Server) 或 [IIS](https://zh.wikipedia.org/wiki/IIS" \o "IIS)，作为独立服务器执行。

目前，Node.js 已被[IBM](https://zh.wikipedia.org/wiki/IBM" \o "IBM)、[Microsoft](https://zh.wikipedia.org/wiki/Microsoft" \o "Microsoft)、[Yahoo!](https://zh.wikipedia.org/wiki/Yahoo!" \o "Yahoo!)、[Walmart](https://zh.wikipedia.org/wiki/Walmart" \o "Walmart)、[Groupon](https://zh.wikipedia.org/wiki/Groupon" \o "Groupon)、[SAP](https://zh.wikipedia.org/wiki/SAP_SE" \o "SAP SE)、[LinkedIn](https://zh.wikipedia.org/wiki/LinkedIn" \o "LinkedIn)、[Rakuten](https://zh.wikipedia.org/wiki/Rakuten" \o "Rakuten)、[PayPal](https://zh.wikipedia.org/wiki/PayPal" \o "PayPal)、[Voxer](https://zh.wikipedia.org/w/index.php?title=Voxer&action=edit&redlink=1" \o "Voxer（页面不存在）)、[GoDaddy](https://zh.wikipedia.org/wiki/GoDaddy" \o "GoDaddy)等企业采用。

Node.js可透过[JavaScript](https://zh.wikipedia.org/wiki/JavaScript" \o "JavaScript)和一系列模块来编写[服务器](https://zh.wikipedia.org/wiki/%E6%9C%8D%E5%8A%A1%E5%99%A8" \o "服务器)端应用和网络相关的应用。核心模块包括[文件系统](https://zh.wikipedia.org/wiki/%E6%96%87%E4%BB%B6%E7%B3%BB%E7%BB%9F" \o "文件系统)I/O、网络（[HTTP](https://zh.wikipedia.org/wiki/HTTP" \o "HTTP)、[TCP](https://zh.wikipedia.org/wiki/%E4%BC%A0%E8%BE%93%E6%8E%A7%E5%88%B6%E5%8D%8F%E8%AE%AE" \o "传输控制协议)、[UDP](https://zh.wikipedia.org/wiki/%E7%94%A8%E6%88%B7%E6%95%B0%E6%8D%AE%E6%8A%A5%E5%8D%8F%E8%AE%AE" \o "用户数据报协议)、[DNS](https://zh.wikipedia.org/wiki/DNS" \o "DNS)、[TLS/SSL](https://zh.wikipedia.org/wiki/%E5%82%B3%E8%BC%B8%E5%B1%A4%E5%AE%89%E5%85%A8%E5%8D%94%E8%AD%B0" \o "传输层安全协议)等）、[二进制](https://zh.wikipedia.org/wiki/%E4%BA%8C%E8%BF%9B%E5%88%B6" \o "二进制)数据流、[加密](https://zh.wikipedia.org/wiki/%E5%8A%A0%E5%AF%86" \o "加密)算法、[数据流](https://zh.wikipedia.org/wiki/%E6%95%B0%E6%8D%AE%E6%B5%81" \o "数据流)等等。Node模块的API形式简单，降低了编程的复杂度。

使用框架可以加速开发。常用的框架有[Express.js](https://zh.wikipedia.org/wiki/Express.js" \o "Express.js)、[Socket.IO](https://zh.wikipedia.org/wiki/Socket.IO" \o "Socket.IO)和Connect等。Node.js的程序可以在[Microsoft Windows](https://zh.wikipedia.org/wiki/Microsoft_Windows" \o "Microsoft Windows)、[Linux](https://zh.wikipedia.org/wiki/Linux" \o "Linux)、[Unix](https://zh.wikipedia.org/wiki/Unix" \o "Unix)、[Mac OS X](https://zh.wikipedia.org/wiki/Mac_OS_X" \o "Mac OS X)等服务器上运行。Node.js也可以使用[CoffeeScript](https://zh.wikipedia.org/wiki/CoffeeScript" \o "CoffeeScript)（一种旨在简化JavaScript的替代语言，其代码可按照一定规则转化为合法的JavaScript代码）、[TypeScript](https://zh.wikipedia.org/wiki/TypeScript" \o "TypeScript)（微软开发的强化了数据类型的JavaScript变体）、[Dart](https://zh.wikipedia.org/wiki/Dart" \o "Dart)语言，以及其他能够编译成JavaScript的语言编程。

Node.js主要用于编写像Web服务器一样的网络应用，这和[PHP](https://zh.wikipedia.org/wiki/PHP" \o "PHP)和[Python](https://zh.wikipedia.org/wiki/Python" \o "Python)是类似的。但是Node.js与其他语言最大的不同之处在于，PHP等语言是阻塞的（只有前一条命令执行完毕才会执行后面的命令），而Node.js是非阻塞的（多条命令可以同时被运行，通过[回调函数](https://zh.wikipedia.org/wiki/%E5%9B%9E%E8%B0%83%E5%87%BD%E6%95%B0" \o "回调函数)得知命令已结束运行）。

Node.js是[事件驱动](https://zh.wikipedia.org/wiki/%E4%BA%8B%E4%BB%B6%E9%A9%85%E5%8B%95" \o "事件驱动)的。开发者可以在不使用[线程](https://zh.wikipedia.org/wiki/%E7%BA%BF%E7%A8%8B" \o "线程)的情况下开发出一个能够承载高并发的服务器。其他服务器端语言难以开发高并发应用，而且即使开发出来，性能也不尽如人意。Node.js正是在这个前提下被创造出来。Node.js把JavaScript的易学易用和Unix网络编程的强大结合起来。

Node.js使用[Google](https://zh.wikipedia.org/wiki/Google" \o "Google) [V8 JavaScript引擎](https://zh.wikipedia.org/wiki/V8_(JavaScript%E5%BC%95%E6%93%8E)" \o "V8 (JavaScript引擎))，因为：

V8是基于[BSD许可证](https://zh.wikipedia.org/wiki/BSD%E8%AE%B8%E5%8F%AF%E8%AF%81" \o "BSD许可证)的开源软件

V8速度非常快

V8专注于网络功能，在HTTP、DNS、TCP等方面更加成熟

Node.js已经有数十万模块，它们可以透过一个名为[npm](https://zh.wikipedia.org/wiki/Npm" \o "Npm)的管理器免费下载。

##### 2.3.1核心模块

在Node中，模块分两类：一类是Node提供的模块，成为核心模块；另一类是用户编写的模块，成为文件模块。核心模块部分在Node源代码的编译过程中，编译进了二进制执行文件。在Node进程启动时，部分核心模块就被直接加载进内存中，所以这部分核心模块引入时，文件定位和编译执行这两个步骤可以省略掉，并且在路径分析中优先判断，所以它的加载速度是最快的。文件模块则是在运行时动态加载，需要完整的路径分析、文件定位、编译执行过程，速度比核心模块慢。

##### 2.3.2异步I/O

在Node中，绝大多数的操作都以异步的方式调用。作者Ryan Dahl排除万难，在底层构建了很多异步I/O的API，从文件读取到网络请求等，均是如此。这样的意义在于，在Node中，我们可以从语言层面很自然的进行并行I/O操作。每个调用之间无须等待之前的I/O调用结束。在编程模型上可以极大提升效率。而对于同步I/O而言，它们的耗时是两个任务的耗时之和。这里异步带来的优势是显而易见的。

##### 2.3.3事件与回调函数

无论在前端还是后端，事件都是常用的。对于其他语言来说，这种俯拾都是JavaScript的熟悉感觉是基本不会出现的。事件的编程方式具有轻量级、松耦合、只关注事务点等优势，但是在多个异步任务的场景下，事件与事件之间各自独立，如何协作是一个问题。

在JavaScript中，回调函数无处不在，这是因为在JavaScript中，我们将函数作为第一等公民来对待，可以将函数作为对象传递给方法作为实参进行调用。

与其他Web后端编程语言相比，Node除了异步和事件之外，回调函数是一大特色。纵观下来，回调函数也是最好的接受异步调用返回数据的方式。

##### 2.3.4单线程

Node保持了JavaScript在浏览器中单线程的特点。而且在Node中，JavaScript与其余线程是无法共享任何状态的。单线程的最大好处是不用像多线程那样处处在意状态的同步问题，此处没有死锁的存在，也没有线程上下文交换所带来的性能上的开销。

同时，单线程也有它自身的弱点，具体在于：1.无法利用多核CPU；2.错误会引起整个应用退出，应用的健壮性值得考验；3.大量计算占用CPU导致无法继续调用异步I/O。

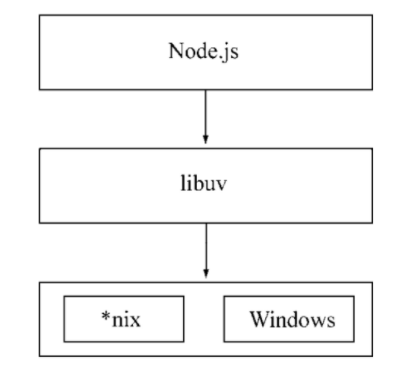
对此，Node采用子进程来解决单线程中大计算量的问题，子进程的出现意味着Node可以从容地应对单线程在健壮性和无法利用多核CPU方面的问题。

##### 2.3.5跨平台

起初Node只支持Linux平台运行，在Node v0.6.0版本发布时，Node已经能够直接在Windows平台上运行了。

兼容Windows和\*nix平台主要得益于Node在架构层面的改动，它在操作系统与Node上层模块系统之间构建了一层平台层架构，即libuv。目前libuv已经成为许多系统实现跨平台的基础组件。

通过良好的的架构，Node的第三方C++模块也可以借助libuv实现跨平台。跨平台架构如图所示。



### 2.4 Express框架概述

Express.js或简称Express，是针对[Node.js](https://zh.wikipedia.org/wiki/Node.js" \o "Node.js)的[web应用框架](https://zh.wikipedia.org/wiki/Web%E5%BA%94%E7%94%A8%E6%A1%86%E6%9E%B6" \o "Web应用框架)，在[MIT许可证](https://zh.wikipedia.org/wiki/MIT%E8%AE%B8%E5%8F%AF%E8%AF%81" \o "MIT许可证)下作为[自由及开放源代码软件](https://zh.wikipedia.org/wiki/%E8%87%AA%E7%94%B1%E5%8F%8A%E5%BC%80%E6%94%BE%E6%BA%90%E4%BB%A3%E7%A0%81%E8%BD%AF%E4%BB%B6" \o "自由及开放源代码软件)发行。它设计用来建造[web应用](https://zh.wikipedia.org/wiki/%E7%BD%91%E7%BB%9C%E5%BA%94%E7%94%A8%E7%A8%8B%E5%BA%8F" \o "网络应用程序)和[API](https://zh.wikipedia.org/wiki/API" \o "API)。它已经被称为针对Node.js的服务器框架的[事实标准](https://zh.wikipedia.org/wiki/%E4%BA%8B%E5%AF%A6%E6%A8%99%E6%BA%96" \o "事实标准)。Express 不对 node.js 已有的特性进行二次抽象，只是在它之上扩展了Web应用所需的功能。丰富的HTTP工具以及来自Connect框架的中间件随取随用，创建强健、友好的API变得快速又简单。

Express 框架核心特性：

* 可以设置中间件来响应 HTTP 请求。
* 定义了路由表用于执行不同的 HTTP 请求动作。
* 可以通过向模板传递参数来动态渲染 HTML 页面。

### 2.5低代码平台概述

低代码是基于可视化和模型驱动理念，结合云原生与多端体验技术，它能够在多数业务场景下实现大幅度的提高工作效率降低企业成本，为专业开发者提供了一种全新的高生产力开发范式。 另一方面，低代码能够让不懂代码的人，通过“拖拉拽”开发组件，就能完成应用程序的搭建。 从意义上讲，低代码可以弥补日益扩大的专业技术人才缺口，同时促成业务与技术深度协作的终极敏捷形态。

可视化表单建模是低代码/零代码平台的核心功能，业内对该功能有多种叫法：电子表单、表单可视化、表单驱动、表单引擎等，该组件主要由表单设计器、表单解析引擎、表单存储引擎三个部分构成，而表单解析引擎取决于表单存储引擎的技术方案。

##### 2.5.1表单设计器

本项目《基于Node.js的台账管理系统》基于K-form-design表单设计包完成表单设计器搭建，k-form-design又基于Vue2和ant-design-vue实现表单设计器，样式使用less作为开发语言，主要功能是通过简单操作来生成配置表单，生成可保存的JSON数据，并将JSON还原成表单，使表单开发更简单更快速。

##### 2.5.2表单解析引擎

本项目使用k-form-build实现表单解析引擎，解析JSON数据生成表单。

##### 2.5.3表单存储引擎

低代码平台需要支持用户存储自定义数据，因为每个应用所需的表、字段、以及关系都是不一样的。自定义数据存储是后端低代码最重要的功能，使用什么方案将直接影响这个产品的适用范围。本项目使用mysql创建表单创建form\_json字段直接存储用户所定义的表单相对应的JSON字符串。

### 2.6技术选型

##### 2.6.1 MySQL

MySQL，原本是一个[开放源码](https://zh.wikipedia.org/wiki/%E9%96%8B%E6%94%BE%E6%BA%90%E7%A2%BC" \o "开放源码)的[关系数据库管理系统](https://zh.wikipedia.org/wiki/%E9%97%9C%E8%81%AF%E5%BC%8F%E8%B3%87%E6%96%99%E5%BA%AB%E7%AE%A1%E7%90%86%E7%B3%BB%E7%B5%B1" \o "关系数据库管理系统)，原开发者为[瑞典](https://zh.wikipedia.org/wiki/%E7%91%9E%E5%85%B8" \o "瑞典)的[MySQL AB](https://zh.wikipedia.org/wiki/MySQL_AB" \o "MySQL AB)公司，该公司于2008年被[昇阳微系统](https://zh.wikipedia.org/wiki/%E6%98%87%E9%99%BD%E5%BE%AE%E7%B3%BB%E7%B5%B1" \o "昇阳微系统)（Sun Microsystems）收购。2009年，[甲骨文公司](https://zh.wikipedia.org/wiki/%E7%94%B2%E9%AA%A8%E6%96%87%E5%85%AC%E5%8F%B8" \o "甲骨文公司)（Oracle）收购昇阳微系统公司，MySQL成为Oracle旗下产品。

MySQL在过去由于性能高、成本低、可靠性好，已经成为最流行的开源数据库，因此被广泛地应用在[Internet](https://zh.wikipedia.org/wiki/Internet" \o "Internet)上的中小型[网站](https://zh.wikipedia.org/wiki/%E7%B6%B2%E7%AB%99" \o "网站)中。随着MySQL的不断成熟，它也逐渐用于更多大规模[网站](https://zh.wikipedia.org/wiki/%E7%BD%91%E7%AB%99" \o "网站)和应用，比如[维基百科](https://zh.wikipedia.org/wiki/%E7%BB%B4%E5%9F%BA%E7%99%BE%E7%A7%91" \o "维基百科)、[Google](https://zh.wikipedia.org/wiki/Google" \o "Google)和[Facebook](https://zh.wikipedia.org/wiki/Facebook" \o "Facebook)等网站。非常流行的开源软件组合[LAMP](https://zh.wikipedia.org/wiki/LAMP" \o "LAMP)中的“M”指的就是MySQL。

但被甲骨文公司收购后，Oracle大幅调涨MySQL商业版的售价，且甲骨文公司不再支持另一个自由软件项目[OpenSolaris](https://zh.wikipedia.org/wiki/OpenSolaris" \o "OpenSolaris)的发展，因此导致自由软件社群们对于Oracle是否还会持续支持MySQL社群版（MySQL之中唯一的免费版本）有所隐忧，MySQL的创始人麦克尔·维德纽斯以MySQL为基础，成立分支计划[MariaDB](https://zh.wikipedia.org/wiki/MariaDB" \o "MariaDB)。而原先一些使用MySQL的开源软件逐渐转向MariaDB或其它的数据库。例如维基百科已于2013年正式宣布将从MySQL迁移到[MariaDB](https://zh.wikipedia.org/wiki/MariaDB" \o "MariaDB)数据库。

与其他的大型数据库例如[Oracle](https://zh.wikipedia.org/wiki/Oracle%E6%95%B0%E6%8D%AE%E5%BA%93" \o "Oracle数据库)、[IBM DB2](https://zh.wikipedia.org/wiki/IBM_DB2" \o "IBM DB2)、[MS SQL](https://zh.wikipedia.org/wiki/Microsoft_SQL_Server" \o "Microsoft SQL Server)等相比，MySQL自有它的不足之处，如规模小、功能有限等，但是这丝毫也没有减少它受欢迎的程度。对于一般的个人用户和中小型企业来说，MySQL提供的功能已经绰绰有余，而且由于MySQL是开放源码软件，因此可以大大降低总体拥有成本。

本项目使用 Navicat 对MySQL数据库进行可视化管理。

##### 2.6.2 Vue

Vue 是一套用于构建用户界面的[开源](https://zh.wikipedia.org/wiki/%E5%BC%80%E6%BA%90%E8%BD%AF%E4%BB%B6" \o "开源软件)[Model–view–viewmodel](https://zh.wikipedia.org/wiki/MVVM" \o "MVVM)[前端](https://zh.wikipedia.org/wiki/%E5%89%8D%E7%AB%AF%E5%92%8C%E5%90%8E%E7%AB%AF" \o "前端和后端)[JavaScript](https://zh.wikipedia.org/wiki/JavaScript" \o "JavaScript)渐进式框架。与其它大型框架不同的是，Vue 被设计为可以自底向上逐层应用。Vue 的核心库只关注视图层，不仅易于上手，还便于与第三方库或既有项目整合。另一方面，当与[现代化的工具链](https://cn.vuejs.org/v2/guide/single-file-components.html)以及各种[支持类库](https://github.com/vuejs/awesome-vue" \l "libraries--plugins" \t "https://cn.vuejs.org/v2/guide/_blank)结合使用时，Vue 也完全能够为复杂的单页应用提供驱动。

##### 2.6.3 Vuex

Vuex 是一个专为 Vue.js 应用程序开发的状态管理模式 + 库。它采用集中式存储管理应用的所有组件的状态，并以相应的规则保证状态以一种可预测的方式发生变化。

##### 2.6.4 element-ui

Element，一套为开发者、设计师和产品经理准备的基于 Vue 2.0 的桌面端组件库。基于四大设计原则：一致性、反馈、效率、可控。使用组件 Demo 快速体验交互细节；使用前端框架封装的代码帮助工程师快速开发。可在线主题编辑器，可视化定制和管理站点主题、组件样式。能下载相关资源，用其快速搭建页面原型或高保真视觉稿，提升产品设计效率。

##### 2.6.5 axios

Axios，是一个基于promise网络请求库，作用于[node.js](https://baike.baidu.com/item/node.js/7567977" \t "https://baike.baidu.com/item/axios/_blank)和[浏览器](https://baike.baidu.com/item/%E6%B5%8F%E8%A7%88%E5%99%A8/213911" \t "https://baike.baidu.com/item/axios/_blank)中，它是 isomorphic 的(即同一套代码可以运行在浏览器和node.js中)。在[服务端](https://baike.baidu.com/item/%E6%9C%8D%E5%8A%A1%E7%AB%AF/6492316" \t "https://baike.baidu.com/item/axios/_blank)它使用原生node.js [http](https://baike.baidu.com/item/http/243074" \t "https://baike.baidu.com/item/axios/_blank)模块, 而在客户端 (浏览端) 则使用XMLHttpRequests。axios本质上也是对原生XHR的封装，只不过它是Promise的实现版本，符合最新的ES规范。

##### 2.6.6 JSON WEB TOKEN（JWT）

JSON Web Token（JWT）是一个轻量级的认证规范，这个规范允许我们使用JWT在用户和服务器之间传递安全可靠的信息。其本质是一个token，是一种紧凑的URL安全方法，用于在网络通信的双方之间传递。

### 2.7本章小结

本项目后端使用JWT实现登录token生成，确保安全通信。通过MySQL关系型数据库存储数据，基于SQL语言对数据进行增删改查操作。前端使用Vue2全家桶构建。基于Element-UI组件库完成前端页面搭建。基于Axios实现异步请求。

# 系统分析与设计

### 3.1系统的需求分析

##### 3.1.1系统功能需求

##### 3.1.2系统性能需求

### 3.2系统功能模块设计

##### 3.2.1管理员模块

3.2.1.1用户管理

3.2.1.2角色管理

3.2.1.3字典管理

3.2.1.4业务单位管理

3.2.1.5台账模板管理

##### 3.2.2管理单位模块

3.2.2.1台账管理

3.2.2.2台账审批

3.2.2.3归档管理

3.2.2.4材料导出

##### 3.2.3下属单位模块

3.2.3.1材料提交

3.2.3.2材料导出

### 3.3系统流程图

### 3.4数据库设计

##### 3.4.1数据库命名规则

##### 3.4.2数据库表结构

### 3.5本章小结

# 系统实现与测试

### 4.1开发环境

### 4.2系统模块实现

##### 4.2.1管理员模块

##### 4.2.2管理单位模块

##### 4.2.3下属单位模块

### 4.3系统测试

##### 4.3.1测试环境

##### 4.3.2测试方法

##### 4.3.3测试结果

### 4.4本章小结

# 总结与展望

### 5.1总结

### 5.2展望

# 参考文献

# 作者简介

# 致谢