

**本科毕业设计论文**

**城市配送WEB前端系统设计**

Design of WEB front end system for urban distribution

专业名称 网络工程

学生姓名 方琴

学生学号 1308093045

指导教师 程博良

毕业时间 2017.7

目 录

[摘 要 4](#_Toc16743)

[致 谢 6](#_Toc15081)

[声 明 7](#_Toc26113)

[正 文 8](#_Toc29649)

[1. 绪论 8](#_Toc23363)

[2. 可行性分析 8](#_Toc30646)

[2.1. 技术可行性 8](#_Toc22966)

[2.2. 操作可行性 8](#_Toc16032)

[2.3. 市场可行性 9](#_Toc7252)

[3. 系统需求分析 9](#_Toc5710)

[3.1. 系统目标 9](#_Toc1487)

[3.2. 系统总体功能描述 9](#_Toc6897)

[3.3. 系统结构图 10](#_Toc31948)

[4. 技术介绍 10](#_Toc20203)

[4.1. webpack 10](#_Toc7983)

[4.2. angularJS 10](#_Toc4463)

[4.3. ui-router 10](#_Toc11448)

[4.4. ui-grid 10](#_Toc29489)

[5. 环境搭建 11](#_Toc19317)

[5.1. 安装nodeJs 11](#_Toc8587)

[5.2. 安装webpack 11](#_Toc7384)

[5.3. 打包配置文件 11](#_Toc9143)

[6. 组件结构 12](#_Toc4212)

[7. 实际开发 12](#_Toc17182)

[7.1. 定义组件容器 12](#_Toc2478)

[7.2. 入口文件 13](#_Toc32452)

[7.3. 头部组件 14](#_Toc22500)

[7.4. 主导航组件 15](#_Toc11088)

[7.5. 首页组件 18](#_Toc15573)

[7.6. 数据列表组件 19](#_Toc11350)

[7.7. 会员中心组件 22](#_Toc23382)

[7.8. 登录组件 22](#_Toc22714)

[7.9. 版权信息组件 23](#_Toc30868)

[7.10. 定义异步加载服务 23](#_Toc27497)

[8. 问题剖析 24](#_Toc3182)

[8.1. 异步加载数据成功后不能立即同步页面刷新 24](#_Toc10951)

[8.2. 使用window.location.href切换页面不成功 25](#_Toc16819)

[8.3. 一次对多个ui-view插入组件时不可以对组件传参数 25](#_Toc1141)

[9. 系统运行 25](#_Toc19788)

[9.1. 项目打包运行 25](#_Toc2144)

[9.2. 浏览器访问 26](#_Toc31629)

[结 论 30](#_Toc17721)

[参考文献 31](#_Toc3403)

# 摘 要

在时代的发展下随着计算机网络技术飞速发展、物流信息管理科学化步伐飞驰不断加快，网络化管理越来越流行已成为一种趋势。如今的物流信息管理受非常多的因素影响，使得物流管理水平相应得不到提高。所以，物流企业被驱使的必须要加强本身综合竞争力和核心竞争力，采用计算机网络技术来加强物流信息的管理，积极实施和物流相关行业未来的发展战略。

这个项目主要是做南昌市的城市配送，业务方面主要是展示在线交易，城配资源，物流相关和物流数据。功能方面主要是实现信息的增删改查。技术方面采用webpack打包工具和angularJs作为基础框架，导航使用ui-router实现单页面应用导航，数据列表采用ui-grid插件，样式风格则采用angular-bootstrap插件。头部和导航栏都封装成组件，主要内容区采用几个组件，类似的模板采用同一个插件，根据参数来配置页面。

关键词

城市配送、angularJs、webpack、ui-router、ui-grid、angular-bootstrap。

Abstract

With the development of computer network technology and the rapid development of logistics information management, network management has become a trend. Logistics information management is affected by many factors, so that the level of logistics management can not be improved. Therefore, the logistics enterprises must enhance their core competitiveness, the use of computer network technology to strengthen logistics information management, and actively implement the logistics industry in the future development strategy.

This project is mainly to do the city of Nanchang City, the main business is to display online transactions, the city with resources, logistics and logistics related data. The main function is to achieve information crud. Technology using webpack package tool and angularJs as a framework, the use of ui-router to achieve a single page navigation navigation application, using ui-grid plug-in data list, using angular-bootstrap plug-in style. The head and navigation bar are packaged into components, the main content area using a few components, similar to the same template with a plug-in, according to the parameters to configure the page.

Keywords

urban distribution、 angularJs、 webpack、 ui-router、 ui-grid、 angular-bootstrap.

# 致 谢

首先感谢我的导师程老师在项目开发时环境搭建与开发技术的选择等方面给了我很多有建设性的建议和帮助，当然还有就是我的同学们，是他们帮助了我更深刻的理解HTML语言，还有就是那些帮助过我的不知姓名的网友，不厌烦的一次又一次的帮我解答问题，没有大家的帮助，我是不可能完成这次设计的。一个人的力量是有限的，发现问题的角度也有限，解决问题的方法也是单一的，非常感谢同学们和我对项目的沟通与建议，让我思维更加开阔，丰富，全面。也非常感谢在大学期间对我们尽心尽力的老师们，你们辛勤的劳动让我们在大学期间收货了人生中宝贵的知识财富，让我们能勇敢的面对走出社会的人生，谢谢你们不求回报的付出，谢谢你们倾囊相授的帮助。

# 声 明

本人郑重声明：

所呈交的毕业设计（论文）是本人在指导教师指导下进行的研究讨论工作及取得的研究成果。其中除加了以标注和致谢的地方，以及法律规定允许的之外，不包含任何人已经撰写或发表完成并以各种方式公开发表过的研究成果，也不包含为获得其他教育机构的学位或证书而作的材料。其他朋友在项目中对本次开发所做的任何帮助与贡献都已经在本文中作了明确的说明并且表示谢意。

本毕业设计（论文）成果是本人在江西师范大学读书期间在指导教师指导下取得的，成果归江西师范大学所有。

特此声明。

声明人（毕业设计（论文）作者）学号：1308093045

声明人（毕业设计（论文）作者）签名：方琴

签名日期：　　2017年　5　月　4　日

# 正 文

1. 绪论

前端发展迅速，框架不断的在更新换代，现在比较流行的就是angularJs，reactJs，vue.js等MVVM开发模式的框架，作为一个前端coder，我们需要与时俱进，紧跟前端发展的步伐，所以我就从angularJs开始学习，然后通过完成城市配送这个前端项目探索如何更深入的了解angularJs，如何独立思考解决问题。

在学习angulaJS的过程中我完全颠覆了原来的开发思想，angularJs呈现页面数据采用双向绑定，页面上数据的改变会及时更新到控制器的域$scope对象上，对$scope属性的改变也会及时渲染到页面上。通过这种方法实现数据表格呈现时采用原生的表格对数据进行循环虽然可以实现数据的呈现，但是可读性不高，灵活性比较差。所以我采用ui-grid插件来呈现表格数据，通过对ui-grid进行简单的配置就可以实现非常灵活的表格功能，并且可读性非常高。

这是我第一次接触单页面应用，使用ui-route能实现公用同一个公共模块，然后基于ui-sref的参数切换使得ui-view现实区域显示不同的模块，这样就不用每次点击导航后进行页面跳转，并且每个页面都得引入公共模块。

1. 可行性分析

项目开始时对可行性分析是很有必要且非常重要，可行性分析的主要目的是我们对项目的实用性进行分析与考察，根据项目的使用范围和使用对象进行分析，考察项目的实用性，可行性。我们主要从三个方面讨论这个项目的可行性，技术可行性，操作可行性和市场可行性。

* 1. 技术可行性

这个项目主要采用angularJS和它的一系列插件相辅相成的开发，angularJs采用灵活的各种指令供应开发过程中的需求，并且可以自定义组件实现想要的功能。angularJs通过对组件的封装使得代码的复用性很高。ui-route实现浏览器端的页面切换，并且可以传递参数，可以不同页面调用不同的组件，也可以不同页面调用同一个组件，通过传递的参数呈现类似的但是数据不同的页面，这个项目非常多的用到了同一个组件呈现不同的页面，代码效率非常高。

* 1. 操作可行性

项目的功能主要是将城市的配送资源整合发布出来，方便大家共享信息共享需求，能够最大化的利用城市的资源，给彼此一个更大的交互平台，对平台上数据的分析可以给用户提供更加合适的方案和更多的参考方案。

* 1. 市场可行性

这个项目旨在给用户提供良好的可视化信息共享平台，给物流资源提供发布平台，用户可以在平台发布货源信息，货源需求，车源信息，车源需求，还可以发布招聘信息，相关人才也可以让大家看见自己。主要是在南昌市内部进行物流相关的各种业务。

1. 系统需求分析

为了能够合理快速的开发必须对城市配送的业务有些了解，城市配送的需求包括物流相关信息的展示，政策公告的宣传，最新货源与车源的展示与发布。还有对城市资源例如仓库资源，设备资源的需求与供应建立连接平台，使得交易更加方便快捷。系统还提供了与物流相关的物流人才与物流招聘。只有对这些业务内容做好准备工作与需求分析，才是承接系统的关键阶段。

* 1. 系统目标

这个系统的开发目标是打造城市配送一体化，平台化，快捷高效的配送模式，但是这个目标过于庞大，不是我一个人能够完成的，作为我个人来说，我的目标是通过这个系统学习angularJs和与其搭配使用的一系列插件。在开发中能独立解决各种问题，思考出最佳的方案，提升自己的学习能力与技术能力。

* 1. 系统总体功能描述

系统总体功能包括物流相关信息的展示，政策公告的宣传，最新货源与车源的展示与发布，城市资源例如仓库资源，设备资源的需求与供应，物流相关的物流人才与物流招聘信息展示。登录跳转功能，会员中心个人物流信息的发布。

* 1. 系统结构图

个人中心

交易管理

物流相关

城配资源

账号管理

首页

在线交易

物流相关

城配资源

政务资讯

1. 技术介绍
   1. webpack

Webpack是模块打包机，它做的事情是，分析你的项目结构，找到JavaScript模块以及其它的一些浏览器不能直接运行的拓展语言（Scss，TypeScrip等），并将其打包为合适的格式以供浏览器使用。

* 1. angularJS

AngularJS诞生于2009年，由Misko Hevery 等人创建，后为Google所收购。是一款优秀的前端JS框架，已经被用于Google的多款产品当中。AngularJS有着诸多特性，最为核心的是：MVVM、模块化、自动化双向数据绑定、语义化标签、依赖注入等等

* 1. ui-router

UI-router是采用的是一种状态管理机制，“状态”可以继承，“状态”不 禁包含url，还有意见，控制器等，以此来组织路由和控制界面UI的渲染， 而不是单纯的改变路由。对于UI方面提供嵌套视图，可以一个页面嵌套多 个视图，多视图再嵌套单个视图，每个视图可提供其特定的控制器等针对管 理，方便视图重用和功能精细，可以打造十分复杂的web应用程序。

* 1. ui-grid

ui-grid是用于实现数据双向绑定实现数据表格的控件，可以通过ui-grid属性配置灵活的表格，可以加上ui-grid-edit ui-grid-pagination ui-grid-selection ui-grid-exporter ui-grid-resize-columns ui-grid-auto-resize等属性丰富表格。

1. 环境搭建
   1. 安装nodeJs

下载nodejs安装

* 1. 安装webpack

启动cmd键入命令：npm install webpack -g

* 1. 打包配置文件

引入需要的包

var webpack = require(‘webpack’),

Path = require(‘path’);

module.exports = {

配置路口文件的路径

entry:{

Index:[‘webpack/hot/dev-server’,’./src/js/main/index.js’]

},

配置输出打包文件的路径

output:{

publicPath:”http://127.0.0.1:8080/dist/js/main”,

Path:path.resolve(\_dirname,”dist/js/main”),

Filename:”bundle.js”

},

配置解析特殊文件的加载器

module:{

{test:/\.js$/,loader:’babel’,query:{persets:[“es2015”]}},

{test:/\.css$/,loader:’style-loader!css-loader’},

{test:/\.html$/,loader:’html’},

{test:/\.sass$/,loader:’style!css!sass?sourceMap’},

{test:/\.scss$/,loader:’style!css!sass?sourceMap’},

{test:/\.(png|jpg)$/,loader:’url-loader?Limit=8192’}

},

resolve:{

root:’’,

配置默认后缀名

extensions:[‘’,’.js’,’.json’,’.css’,’.sass’],

alias:{}

}

}

1. 组件结构

ui-view

csps-head

csps-table

csps-center

csps-footer

csps-home

csps-login

1. 实际开发
   1. 定义组件容器

<div id="headWrapper">

<div ui-view="head">

<csps-head></csps-head>

</div>

</div>

<div id="viewWrapper">

<div ui-view>

<csps-home></csps-home>

</div>

<div ui-view="c\_content" style="position:absolute;top:0;left:215px;">

</div>

</div>

<div id="footerWrapper">

<csps-footer></csps-footer>

</div>

* 1. 入口文件

入口文件是项目打包时的起始文件，主要是配置主页面的路由，引入包含各个组件的模板文件封装成一个容器组件，用于在index.html页面调用。

定义csps组件，引入需要的ui.router插件，组件和服务。

let csps\_web = angular.module("csps",[

ComponentsModule.name,

ServiceModule.name,

'ui.router',

]).component('app',indexComponent);

设置路由。

csps\_web.config(function($stateProvider) {

var states = [{

定义路由标识

name: 'level\_explain',

url: '/level\_explain',

在该路由下插入ui-view的组件

component : 'cspsTable',

传入组件的参数

resolve: {

option : function(){

let option = {

href : "",

title:"诚信等级说明",

group:"诚信认证",

gridOption : {

paginationPageSizes: [10, 20, 30],

paginationPageSize: 10,

}

}

return option;

}

}

}]

states.forEach(function(state) {

$stateProvider.state(state);

});

}

* 1. 头部组件

头部组件主要包含顶部导航和页面首页横幅展示，根据用户的登录状态顶部导航会有不同的显示。

定义头部的控制器文件

import "./head.scss";

import template from "./head.html";

const headComponent = {

template:template,

bindings : {

option : "<",

},

controller:function($scope) {

if(document.cookie.split("=")[1] == "0"){

$scope.login = false;

}else{

$scope.login = true;

}

}

}

export default headComponent;

引入头部组件的控制器与模板的组合

import headComponent from "./head.component";

定义头部组件

const headModule = angular.module("cspsHead",[])

.component("cspsHead",headComponent);

* 1. 主导航组件

主导航组件是通过ui-sref实现单页面路由跳转，通过在ui-view里面插入不同的组件呈现不同的页面。

定义主导航的控制器文件

import template from "./nav.html";

import "./nav.scss";

const navComponent = {

template : template,

bindings :{

},

controller : ($scope)=>{

$scope.navTree = [

{

fname :"政务咨讯",

selected : false,

className: "",

childrens :[

{name : "图文资讯",route:"information"},

{name : "行业信息",route:"news"},

{name : "政策法规",route:"policy"},

]

},

{

fname :"在线交易",

selected : false,

className: "",

childrens :[

{name : "最新货源",route:"cargo"},

{name : "最新车源",route:"car"},

]

},

{

fname :"城配资源",

selected : false,

className: "",

childrens :[

{name : "仓储出租",route:"house\_rent"},

{name : "托盘供应",route:"tray\_supply"},

{name : "设备供应",route:"equip\_supply"},

{name : "设备需求",route:"equip\_demand"}

]

},

{

fname :"物流相关",

selected : false,

className: "",

childrens :[

{name : "物流招标",route:"log\_tender"},

{name : "物流金融",route:"log\_finance"},

{name : "企业招聘",route:"bus\_recruit"},

{name : "物流人才",route:"log\_talent"}

]

},

{

fname :"物流数据",

selected : false,

className: "",

childrens :[

// {name : "平台数据统计",route:"data\_census"},

{name : "运价指数",route:"operate"}

]

},

{

fname :"诚信认证",

selected : false,

className: "",

childrens :[

{name : "企业诚信排行",route:"credit\_rank"},

// {name : "诚信等级说明",route:"level\_explain"}

]

}

];

}

}

export default navComponent;

引入主导航组件的控制器与模板的组合

import navComponent from "./nav.component";

定义主导航组件

const navModule = angular.module("cspsNav",[])

.component("cspsNav",navComponent);

* 1. 首页组件

首页与其他页面风格完全不一样，所以单独封装出来一个组件，首页上主要有用到bootstrap的轮播样式和标签页样式。

定义首页控制器文件

import template from "./home.html";

import "./home.scss";

const homeComponent = {

template : template,

controller : function ($scope,$timeout,ajaxService) {

$scope.tabId = 0;

$scope.tabDataUrl=["./src/js/common/dataBase/news.json","./src/js/common/dataBase/policy.json"];

$scope.tabData = [];

ajaxService.getData($scope.tabDataUrl[0]).then(res=>{

console.log(res);

$timeout(()=>{

$scope.tabData = res;

});

});

$scope.selectTab = id=>{

$scope.tabId = id;

ajaxService.getData($scope.tabDataUrl[id]).then(res=>{

console.log(res);

$timeout(()=>{

$scope.tabData = res;

});

});

}

}

}

export default homeComponent;

引入首页组件的控制器与模板的组合

import homeComponent from "./home.component";

定义首页组件

const homeModule = angular.module("cspsHome",[])

.component("cspsHome",homeComponent);

* 1. 数据列表组件

数据列表组件很多页面都用上了，主要用来放置过滤表单和数据表格，首页多个连接点击后都是跳转到这个组件，但是页面的呈现是通过参数来做差异的。

定义数据列表组件的控制器

import template from "./table.html";

import "./table.scss";

const tableComponent = {

template : template,

bindings : {

option : "<",

},

controller : function ($scope,$uibModal,$timeout,ajaxService) {

let ctrl = this,

option = ctrl.option;

$scope.title = option.title;

$scope.group = option.group;

$scope.filter = option.filter;

$scope.gridOptions = option.gridOptions;

$scope.gridOptions.enableRowHeaderSelection = false ;

$scope.gridOptions.enableHorizontalScrollbar = 0;

$scope.gridOptions.enableVerticalScrollbar = 0;

ajaxService.getData(option.href).then(res=>{

$timeout(()=>{

ctrl.refreshTable(res);

});

});

this.refreshTable = res=>{

console.log(res);

$scope.gridOptions.data = res;

$scope.leftList = res;

}

$scope.add = function(){

var modalInstance = $uibModal.open({

animation: true,

size : "lg",

component: option.modalComponent,

windowClass:"addModal",

resolve: {

}

});

modalInstance.result.then(function (res) {

console.log(res);

}, function (res) {

console.log(res);

});

}

}

}

export default tableComponent;

引入列表组件的控制器与模板的组合

import tableComponent from "./table.component";

定义数据列表组件，并引入表格所需的插件

const tableModule = angular.module("cspsTable",[

'ui.bootstrap',

'ui.grid',

'ui.grid.edit',

'ui.grid.selection',

'ui.grid.cellNav',

'ui.grid.pagination',

])

.component("cspsTable",tableComponent);

* 1. 会员中心组件

会员中心也有一个左侧导航，同样采用ui-router的ui-sref实现页面的切换，由于当页面同时要在两个ui-view里面放入组件时是传不了参数的，所以虽然会员中心的每个页面都很相似但是却共用不了同一个组件，所以我为每个页面封装了一个组件，这样页面也会更加灵活。但是代码量会大很多。

定义会员中心控制器

import template from "./center.html";

import "./center.scss";

const centerComponent = {

template : template,

bindings : {

option : "<",

},

controller : function ($scope,$uibModal) {

}

}

export default centerComponent;

引入会员中心组件的控制器与模板的组合

import centerComponent from "./center.component";

定义会员中心组件

const centerModule = angular.module("cspsCenter",[])

.component("cspsCenter",centerComponent);

* 1. 登录组件

登录组件登录成功就会设置document.cookie = "login=1";页面根据cookie值显示不同的操作按钮，实现登入效果。

定义登录组件控制器

import "./login.scss";

import template from "./login.html";

const loginComponent = {

template : template,

controller: ()=>{

}

}

export default loginComponent;

引入登录组件的控制器与模板的组合

import loginComponent from "./login.component";

定义登录组件

const loginModule = angular.module("cspsLogin",[])

.component("cspsLogin",loginComponent);

* 1. 版权信息组件

版权信息组件只是简单的显示版权信息页脚，将其封装成一个组件可以节省多处调用。

引入版权信息组件的模板和控制器

import footerComponent from "./footer.component";

定义版权信息组件

const footerModule = angular.module("cspsFooter",[])

.component("cspsFooter",footerComponent);

* 1. 定义异步加载服务

class ajaxService {

constructor($http){

this.$http = $http;

};

getData(href,param){

return new Promise(function(resolve,reject){

$.ajax({

type:"get",

data :param,

dataType : 'json',

url : href,

contentType:'application/json;charset="utf-8"',

success : res=>resolve(res),

error : res=>reject(res)

});

});

}

}

export default ajaxService;

1. 问题剖析
   1. 异步加载数据成功后不能立即同步页面刷新

数据加载采用promise实现异步加载

getData(href,param){

return new Promise(function(resolve,reject){

$.ajax({

type:"get",

data :param,

dataType : 'json',

url : href,

contentType:'application/json;charset="utf-8"',

success : res=>resolve(res),

error : res=>reject(res)

});

});

}

getData(option.href).then(res=>{

ctrl.refreshTable(res);

});

refreshTable = res=>{

console.log(res);

$scope.gridOptions.data = res;

$scope.leftList = res;

}

在控制器里调用getData方法返回一个promise对象，通过.then()里的回 调函数拿到异步加载成功时返回的数据，通过refreshTable将数据绑定到 $scope域上，然后通过数据双向绑定实现页面上的数据实时刷新，但是在数 据成功加载的情况下页面并没有实时刷新。

解决办法：在.then的回调函数里套一层$timeout()在加载数据后延迟几秒 调更新$scope域函数。

getData(option.href).then(res=>{

$timeout(()=>{

ctrl.refreshTable(res);

});

});

* 1. 使用window.location.href切换页面不成功

点击登录时使用window.location.href跳转到登入页面时成功，但是输入用户名和密码登录时通过window.location.href跳转到首页不成功。

我采用另一种方案实现页面的跳转，我采用ui-sref实现浏览器页面跳转，通过设置cookie判断当前状态是否登录。

* 1. 一次对多个ui-view插入组件时不可以对组件传参数

views:{

"":{

component : 'cspsLogin',

},

"head":{

component : 'cspsHead',

}

}

在这种情况下用resolve:{}传参数到组件里是不成功的。

1. 系统运行
   1. 项目打包运行

进入项目目录运行npm install下载相应的工具包，执行webpack-dev-server打包项目。

* 1. 浏览器访问

在浏览器输入网址localhost:8080//webpack-dev-server/index.html进入首页，点击主导航链接可以单页面跳转到多个模块。



点击在线交易-最新货源



点击在线交易-最新车源



点击城配资源-仓储出租



点击城配资源-托盘供应



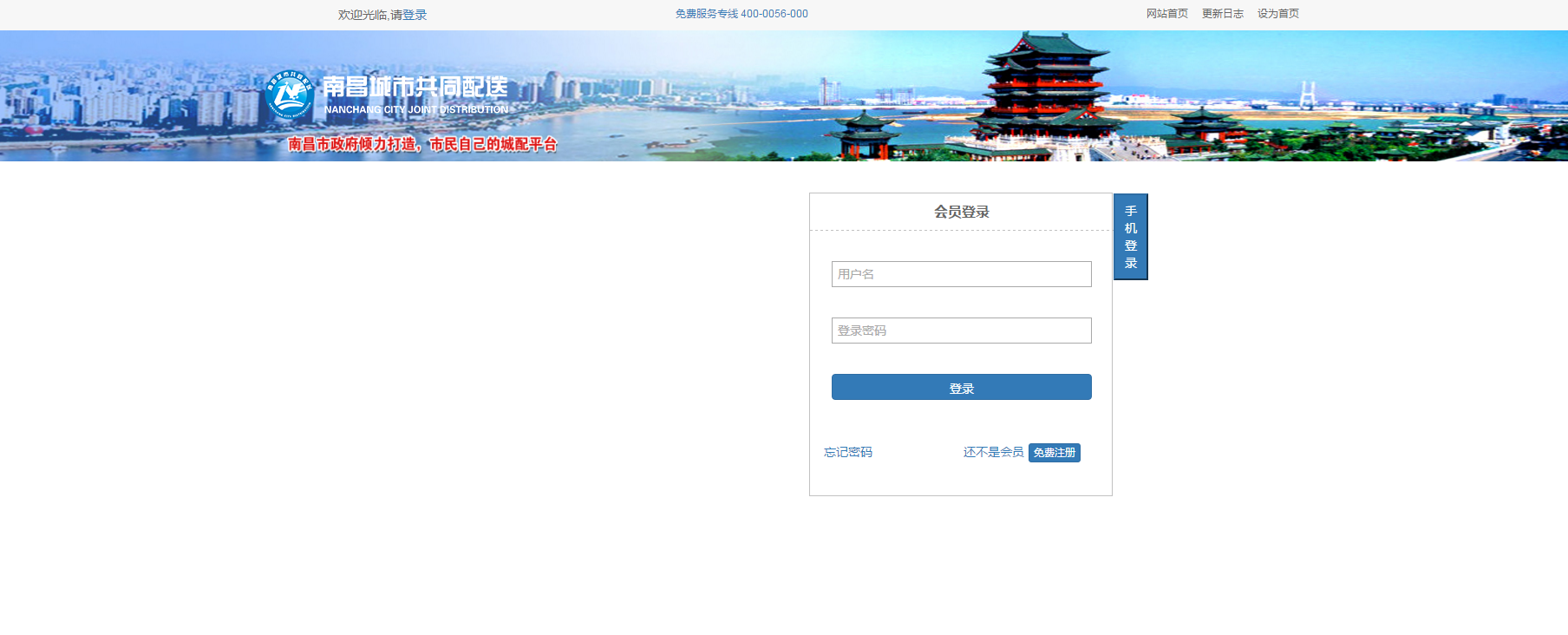
点击城市配送-设备供应



点击城市配送-设备需求



左上角有个登录按钮，点击跳转到登入页面，登入成功跳转回首页，此时页面右上角多了退出和个人中心按钮，点击个人中心进入个人中心。



进入个人中心才可以进行新增修改删除操作。



# 结 论

该系统后端采用spring+springMVC+hibernate框架开发，前端我用angularJS重新开发。之所以选择angularJS是因为现在前端开发发展迅速，研究一种MVVM框架可以转换之前对MVC模式开发的固化思想，开拓自己的知识领域，能更快的接受新知识，跟上前端飞速发展的步伐。在边学习边开发的过程中遇到不少困难，时而感觉很迷茫时而感觉到突破的喜悦，在无数次翻阅资料，Google中提升了自己的自学能力。

ES6的表面上感觉跟ES5除了表现形式的差别外没差太多，但是实际开发中发现还是差别蛮大的，ES6的设计比较适合组件化的开发。例如箭头函数中的this指向的是定义函数所在的类，而不是应用函数的对象，这一点很容易混淆，但是也很重要，换一种说法就是ES6语法中方法没有this这个属性，this变量指向的是定义函数的环境。ES5语法中的变量提升困扰了很多人，在ES5环境中对于变量可以先使用再定义，但是在ES6中没有变量提升，所以要用一个变量时需要先声明。

通过自己独立学习一门框架开发的过程中，发现了学习前端还是蛮有意思的，前端不仅仅只是呈现页面，他也有复杂的逻辑，一个好的前端能够为项目提供高效的入口，为后台提供更方便的需求。

# 参考文献

[1] AngularJs即学即用，（美国）夏德瑞（Shyam Seshadri）、（美国）格林（Brad Green） 译者：朱磊，中国电力出版社，2016

[2] AngularJS深度剖析与最佳实践，机械工业出版社，2015

[3] AngularJs高级程序设计，人民邮电出版社，2015

[4] 深入浅出Node.js，人民邮电出版社，2013

[5] ES6标准入门，电子工业出版社，2015

[6] Adobe Flex 3高级编程，清华大学出版社，2011

[7] CSS高效开发实践，电子工业出版社，2014

[8] HTML&CSS设计与架构网站，清华大学出版社，2013