1.项目概述

1.1.项目演示

1. 运行 “饿了么项目” ，演示应用程序效果，演示 “点餐业务线” 整体流程。
2. 本项目参照 “饿了么官网网页版”制作，演示饿了么官网效果。饿了么网页版：http://h5.ele.me/
3. 本项目专注于完成**点餐业务线**功能， ”饿了么官网“中的其它功能暂不涉及 。

1.2.项目目标

1. 本项目为课程级贯穿项目中的第三个项目（JDBC项目、前端项目、**javaWeb项目**）的服务器端升级版。

   即**数据库结构与前端Vue没有任何变化，而服务器端由Servlet升级为SpringBoot。**
2. 本项目完成后，学员将能够使用VUE+SpringBoot+AJAX技术开发前后端分离的Web应用程序。

1.3.数据库设计

1.3.1.DB一览表

No	表名称	中文名	说明
1	business	商家表	存储所有商家信息
2	food	食品表	存储每个商家所拥有的所有食品信息
3	cart	购物车表	存储每个用户的购物车中的食品信息
4	deliveryaddress	送货地址表	存储每个用户的所有送货地址信息
5	orders	订单表	存储每个用户的所有订单信息
6	orderdetailet	订单明细表	存储每个订单中所订购的所有食品信息
7	user	用户表	存储所有用户信息

1.3.2.表结构

约束类型标识： PK：primary key 主键 FK：foreign key 外键 NN：not null 非空 UQ：unique 唯一索引 AI：auto increment 自增长列

1.3.2.1.business（商家表）

No	字段名	数据类型	size	默认值	约束	说明
1	businessId	int			PK、AI、NN	商家编号
2	businessName	varchar	40		NN	商家名称
3	businessAddress	varchar	50			商家地址
4	businessExplain	varchar	40			商家介绍
5	businessImg	mediumtext			NN	商家图片
6	orderTypeId	int			NN	点餐分类： 1：美食、2：早餐、3：跑腿代购、4：汉堡披萨、5：甜品饮品、6：速食简餐、7：地方小吃、8：米粉面馆、9：包子粥铺、10：炸鸡炸串
7	starPrice	decimal	(5,2)	0.00		起送费
8	deliveryPrice	decimal	(5,2)	0.00		配送费
9	remarks	varchar	40			备注

1.3.2.2.food（食品表）

No	字段名	数据类型	size	默认值	约束	说明
1	foodId	int			PK、AI、NN	食品编号
2	foodName	varchar	30		NN	食品名称
3	foodExplain	varchar	30		NN	食品介绍
4	foodImg	mediumtext			NN	食品图片
5	foodPrice	decimal	(5,2)		NN	食品价格
6	businessId	int			FK、NN	所属商家编号
7	remarks	varchar	40			备注

1.3.2.3.cart（购物车表）

No	字段名	数据类型	size	默认值	约束	说明
1	cartId	int			PK、AI、NN	无意义编号
2	foodId	int			FK、NN	食品编号
3	businessId	int			FK、NN	所属商家编号
4	userId	varchar	20		FK、NN	所属用户编号
5	quantity	int			NN	同一类型食品的购买数量

1.3.2.4.deliveryaddress（送货地址表）

No	字段名	数据类型	size	默认值	约束	说明
1	daId	int			PK、AI、NN	送货地址编号
2	contactName	varchar	20		NN	联系人姓名
3	contactSex	int			NN	联系人性别
4	contactTel	varchar	20		NN	联系人电话
5	address	varchar	100		NN	送货地址
6	userId	varchar	20		FK、NN	所属用户编号

1.3.2.5.orders（订单表）

No	字段名	数据类型	size	默认值	约束	说明
1	orderId	int			PK、AI、NN	订单编号
2	userId	varchar	20		FK、NN	所属用户编号
3	businessId	int			FK、NN	所属商家编号
4	orderDate	varchar	20		NN	订购日期
5	orderTotal	decimal	(7,2)	0.00	NN	订单总价
6	daId	int			FK、NN	所属送货地址编号
7	orderState	int		0	NN	订单状态（0：未支付； 1：已支付）

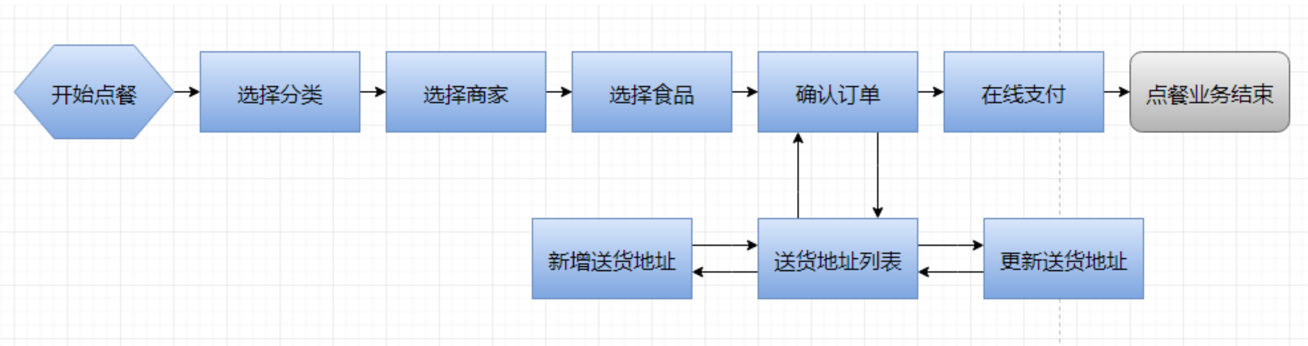
1.3.2.6.orderdetailet（订单明细表）

No	字段名	数据类型	size	默认值	约束	说明
1	odId	int			PK、AI、NN	订单明细编号
2	orderId	int			FK、NN	所属订单编号
3	foodId	int			FK、NN	所属食品编号
4	quantity	int			NN	数量

1.3.2.7.user（用户表）

No	字段名	数据类型	size	默认值	约束	说明
1	userId	varchar	20		PK、NN	用户编号
2	password	varchar	20		NN	密码
3	userName	varchar	20		NN	用户名称
4	userSex	int		1	NN	用户性别（1：男； 0：女）
5	userImg	mediumtext				用户头像
6	delTag	int		1	NN	删除标记（1：正常； 0：删除）

1.4.业务流程



2.服务器端项目搭建

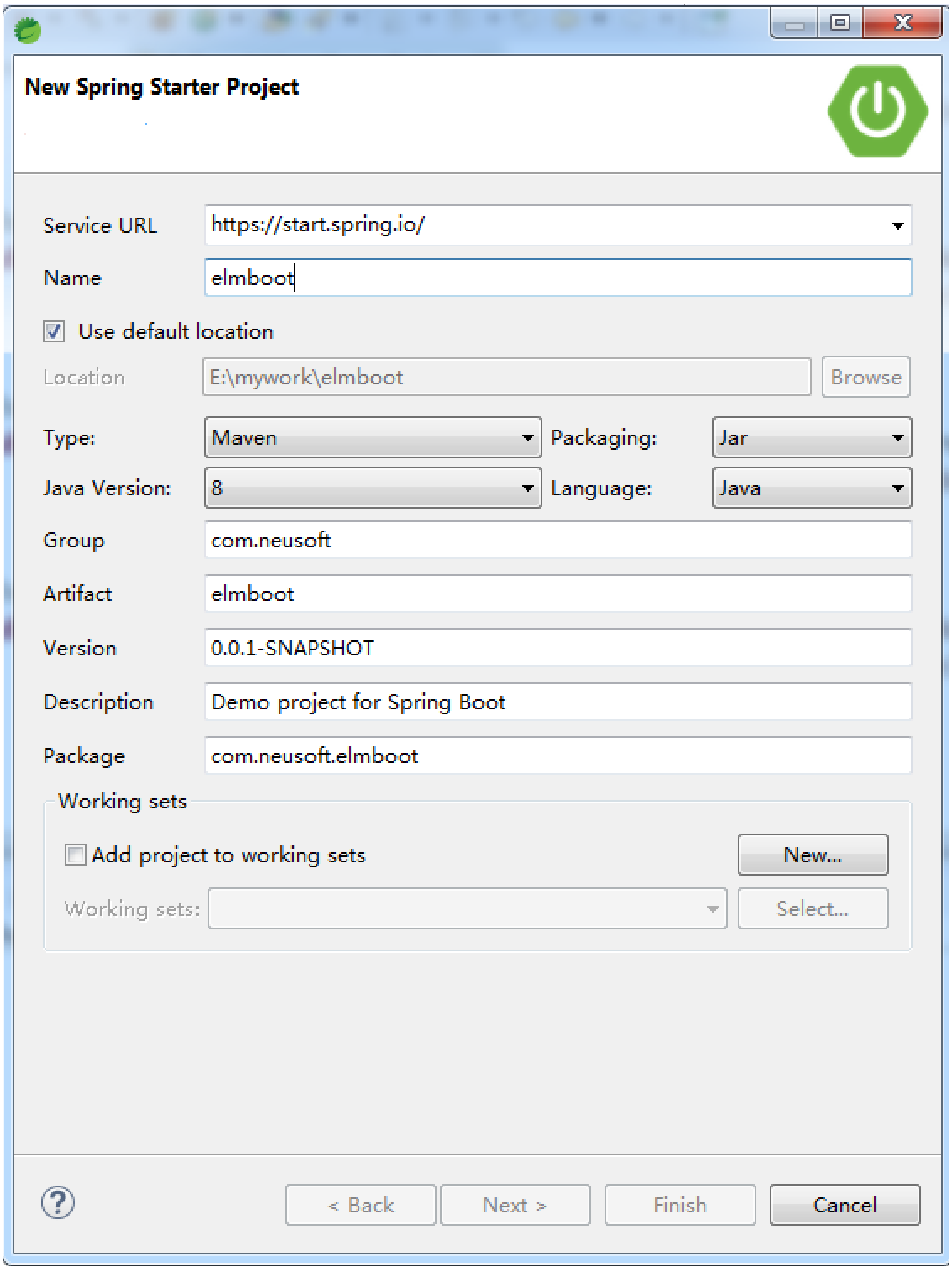
2.1.开发环境检查

1. 开发工具：SpringToolSuite（STS）
2. 检查开发工具的jdk配置：jdk8
3. 检查maven构建工具的配置：maven3
4. 检查开发工具的文件编码配置：utf-8

2.2.搭建springboot工程总体架构

2.2.1.工程类型

创建工程类型：Spring Starter project



New Spring Starter Project

Service URL: https://start.spring.io/

Name: elmboot

☑ Use default location

Location: E:\mywork\elmboot

Type: Maven

Packaging: Jar

Java Version: 8

Language: Java

Group: com.neusoft

Artifact: elmboot

Version: 0.0.1-SNAPSHOT

Description: Demo project for Spring Boot

Package: com.neusoft.elmboot

Working sets

□ Add project to working sets

Working sets:

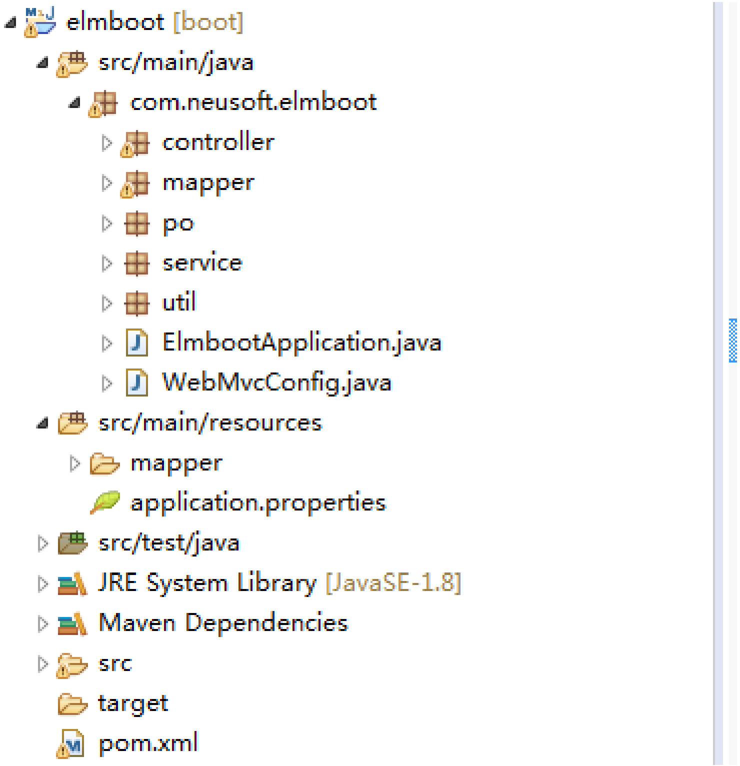
2.2.2.pom.xml文件

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>2.3.0.RELEASE</version>
        <relativePath/> <!-- lookup parent from repository -->
    </parent>
    <groupId>com.neusoft</groupId>
    <artifactId>elmboot</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <name>elmboot</name>
    <description>Demo project for Spring Boot</description>

    <properties>
        <java.version>1.8</java.version>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
        <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
    </properties>

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

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

        <dependency>
            <groupId>org.mybatis.spring.boot</groupId>
            <artifactId>mybatis-spring-boot-starter</artifactId>
            <version>2.0.1</version>
        </dependency>

        <dependency>
            <groupId>mysql</groupId>
            <artifactId>mysql-connector-java</artifactId>
            <version>5.1.6</version>
            <scope>runtime</scope>
        </dependency>

        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
            <scope>test</scope>
            <exclusions>

                <exclusion>
```

```
                    <groupId>org.junit.vintage</groupId>
                    <artifactId>junit-vintage-engine</artifactId>
                </exclusion>
            </exclusions>
        </dependency>
    </dependencies>

    <build>
        <plugins>
            <plugin>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-maven-plugin</artifactId>
            </plugin>
        </plugins>
    </build>

</project>
```

2.2.3.工程目录结构



```
elmboot [boot]
    src/main/java
        com.neusoft.elmboot
            controller
            mapper
            po
            service
            util
            ElmbootApplication.java
            WebMvcConfig.java
    src/main/resources
        mapper
        application.properties
    src/test/java
    JRE System Library [JavaSE-1.8]
    Maven Dependencies
    src
    target
    pom.xml
```

2.2.4.SpringBoot入口文件

```java
package com.neusoft.elmboot;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class ElmbootApplication {

    public static void main(String[] args) {
        SpringApplication.run(ElmbootApplication.class, args);
    }
}
```

2.2.5.处理跨域的Cors配置文件

```java
package com.neusoft.elmboot;

import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.web.servlet.config.annotation.CorsRegistry;
import org.springframework.web.servlet.config.annotation.WebMvcConfigurer;

@Configuration
public class WebMvcConfig {
    @Bean
    public WebMvcConfigurer corsConfigurer() {
        return new WebMvcConfigurer() {
            @Override
            public void addCorsMappings(CorsRegistry registry) {
                registry.addMapping("/**")
                        .allowedOrigins("http://localhost:8081")
                        .allowCredentials(true)
                        .allowedMethods("GET", "POST", "DELETE", "PUT","PATCH")
                        .allowedHeaders("*")
                        .maxAge(36000);
            }
        };
    }
}
```

2.2.6.application.properties配置文件

```
server.port=8080
server.servlet.context-path=/elm

logging.level.org.springframework=debug
logging.level.com.neusoft.elmboot.mapper=debug

spring.datasource.username=root
spring.datasource.password=123
spring.datasource.url=jdbc:mysql://localhost:3306/elm?characterEncoding=utf-8
spring.datasource.driver-class-name=com.mysql.jdbc.Driver

mybatis.mapper-locations=classpath:mapper/*.xml
mybatis.type-aliases-package=com.neusoft.elmboot.po
```

2.3.服务器接口API

2.3.1.business

1. BusinessController/listBusinessByOrderTypeId
   参数：orderTypeId
   返回值：business数组
   功能：根据点餐分类编号查询所属商家信息
2. BusinessController/getBusinessById
   参数：businessId
   返回值：business对象
   功能：根据商家编号查询商家信息

2.3.2.food

1. FoodController/listFoodByBusinessId
   参数：businessId
   返回值：food数组
   功能：根据商家编号查询所属食品信息

2.3.3.cart

1. CartController/listCart
   参数：userId、businessId（可选）
   返回值：cart数组（多对一：所属商家信息、所属食品信息）
   功能：根据用户编号查询此用户所有购物车信息
   根据用户编号和商家编号，查询此用户购物车中某个商家的所有购物车信息
2. CartController/saveCart
   参数：userId、businessId、foodId
   返回值：int（影响的行数）
   功能：向购物车表中添加一条记录

3. CartController/updateCart
   参数：userId、businessId、foodId、quantity
   返回值：int（影响的行数）
   功能：根据用户编号、商家编号、食品编号更新数量
4. CartController/removeCart
   参数：userId、businessId、foodId（可选）
   返回值：int（影响的行数）
   功能：根据用户编号、商家编号、食品编号删除购物车表中的一条食品记录
   根据用户编号、商家编号删除购物车表中的多条条记录

2.3.4.deliveryAddress

1. DeliveryAddressController/listDeliveryAddressByUserId
   参数：userId
   返回值：deliveryAddress数组
   功能：根据用户编号查询所属送货地址
2. DeliveryAddressController/getDeliveryAddressById
   参数：daId
   返回值：deliveryAddress对象
   功能：根据送货地址编号查询送货地址
3. DeliveryAddressController/saveDeliveryAddress
   参数：contactName、contactSex、contactTel、address、userId
   返回值：int（影响的行数）
   功能：向送货地址表中添加一条记录
4. DeliveryAddressController/updateDeliveryAddress
   参数：daId、contactName、contactSex、contactTel、address、userId
   返回值：int（影响的行数）
   功能：根据送货地址编号更新送货地址信息
5. DeliveryAddressController/removeDeliveryAddress
   参数：daId
   返回值：int（影响的行数）
   功能：根据送货地址编号删除一条记录

2.3.5.orders

1. OrdersController/createOrders
   参数：userId、businessId、daId、orderTotal
   返回值：int（订单编号）
   功能：根据用户编号、商家编号、订单总金额、送货地址编号向订单表中添加一条记录，
   并获取自动生成的订单编号，
   然后根据用户编号、商家编号从购物车表中查询所有数据，批量添加到订单明细表中，
   然后根据用户编号、商家编号删除购物车表中的数据。
2. OrdersController/getOrdersById
   参数：orderId
   返回值：orders对象（包括多对一：商家信息；　一对多：订单明细信息）
   功能：根据订单编号查询订单信息，包括所属商家信息，和此订单的所有订单明细信息
3. OrdersController/listOrdersByUserId
   参数：userId

返回值：orders数组（包括多对一：商家信息；　一对多：订单明细信息）
功能：根据用户编号查询此用户的所有订单信息

2.3.6.user

1. UserController/getUserByIdByPass
   参数：userId、password
   返回值：user对象
   功能：根据用户编号与密码查询用户信息
2. UserController/getUserById
   参数：userId
   返回值：int（返回行数）
   功能：根据用户编号查询用户表返回的行数
3. UserController/saveUser
   参数：userId、password、userName、userSex
   返回值：int（影响的行数）
   功能：向用户表中添加一条记录

3.前端项目搭建

3.1.开发环境检查

1. 检查cnpm安装环境： 命令行下输入：cnpm -v
2. 检查VueCli安装环境：命令行下输入：vue -V　（注意：本工程使用VueCli4.0.0以上版本）

> 附录：
>
> 1. 安装npm：直接安装node.js　（输入 "npm -v" 测试是否安装成功； 输入 node –v 查看node版本）
> 2. 安装cnpm：npm install -g cnpm --registry=https://registry.npm.taobao.org
> 3. 全局安装vuecli：cnpm install -g @vue/cli　 （或：npm install -g @vue/cli@4.x.x）
> 4. 查看当前安装的vue-cli版本：vue --version 或 vue –V
> 5. 卸载旧版本的vue-cli：cnpm uninstall vue-cli -g
> 6. 查看远程仓库中的版本号：cnpm view @vue/cli versions --json

3.2.搭建VueCli工程总体架构

3.2.1.搭建VueCli模板工程

1. 命令行下进入工作空间目录中，输入： vue create elmclient　（工程名必须小写）
2. 选择预设模板：这里选择“Manually select features”（手动选择特征）
3. 模块选取：Babel、Router
4. 选择是否使用history 形式的路由：选择：Y
5. 将依赖文件放在package.json中：选择：“in package.json”
6. 是否将当前选择保存以备下次使用：选择：N

7. 进入创建好的工程目录：cd elmclient
8. 启动工程：npm run serve
9. 在浏览器中测试：http://localhost:8080

3.2.2.添加其它依赖及配置文件

1. 添加font-awesome与axios依赖： cnpm install font-awesome --save cnpm install axios --save cnpm install qs --save
2. 添加图片到src的assets中。
3. 在src目录下添加common.js文件

```
//获取当前时间（XXXX-XX-XX)
export function getCurDate() {
    var now = new Date();
    var year = now.getFullYear();
    var month = now.getMonth() + 1;
    var day = now.getDate();
    month = month < 10 ? "0" + month : month;
    day = day < 10 ? "0" + day : day;
    return year + "-" + month + "-" + day;
}

//向sessionStorage中存储一个JSON对象
export function setSessionStorage(keyStr, value) {
    sessionStorage.setItem(keyStr, JSON.stringify(value));
}

//从sessionStorage中获取一个JSON对象（取不到时返回null)
export function getSessionStorage(keyStr) {
    var str = sessionStorage.getItem(keyStr);
    if (str == '' || str == null || str == 'null' || str == undefined) {
        return null;
    } else {
        return JSON.parse(str);
    }
}

//从sessionStorage中移除一个JSON对象
export function removeSessionStorage(keyStr) {
    sessionStorage.removeItem(keyStr);
}

//向localStorage中存储一个JSON对象
export function setLocalStorage(keyStr, value) {
    localStorage.setItem(keyStr, JSON.stringify(value));
}

//从localStorage中获取一个JSON对象（取不到时返回null)
export function getLocalStorage(keyStr) {
    var str = localStorage.getItem(keyStr);
    if (str == '' || str == null || str == 'null' || str == undefined) {
        return null;
    } else {
        return JSON.parse(str);
    }
}

//从localStorage中移除一个JSON对象
export function removeLocalStorage(keyStr) {
    localStorage.removeItem(keyStr);
}
```

4. 在工程根目录下添加vue.config.js文件

```
module.exports = {
    devServer: {
        port: 8081
    }
}
```

3.2.3.main.js文件

```
import Vue from 'vue'
import App from './App.vue'
import router from './router'

import 'font-awesome/css/font-awesome.min.css'
import axios from 'axios'
import qs from 'qs'
import {
    getCurDate,
    setSessionStorage,
    getSessionStorage,
    removeSessionStorage,
    setLocalStorage,
    getLocalStorage,
    removeLocalStorage
} from './common.js'

Vue.config.productionTip = false

//设置axios的基础url部分
axios.defaults.baseURL = 'http://localhost:8080/elm/';
//将axios挂载到vue实例上，使用时就可以 this.$axios 这样使用了
Vue.prototype.$axios = axios;

Vue.prototype.$qs = qs;

Vue.prototype.$getCurDate = getCurDate;
Vue.prototype.$setSessionStorage = setSessionStorage;
Vue.prototype.$getSessionStorage = getSessionStorage;
Vue.prototype.$removeSessionStorage = removeSessionStorage;
Vue.prototype.$setLocalStorage = setLocalStorage;
Vue.prototype.$getLocalStorage = getLocalStorage;
Vue.prototype.$removeLocalStorage = removeLocalStorage;

router.beforeEach(function(to,from,next){
    let user = sessionStorage.getItem('user');
    //除了登录、注册、首页、商家列表、商家信息之外，都需要判断是否登录
    if(!
(to.path=='/'||to.path=='/index'||to.path=='/businessList'||to.path=='/businessInfo'||to.path=='
/login'||to.path=='/register')){
        if(user==null){
            router.push('/login');
            location.reload();
        }
    }
    next();
});

new Vue({
  router,
  render: h => h(App)
}).$mount('#app')
```

3.2.4.App.vue文件

注意：在App.vue文件中，#app的高度也要设置为100%。

```
<template>
    <div id="app">
        <router-view />
    </div>
</template>

<!-- 这里是共通样式，适用于所有组件，所以不要加scoped -->
<style>
    html,body,div,span,h1,h2,h3,h4,h5,h6,ul,ol,li,p {
        margin: 0;
        padding: 0;
    }
    html,body,#app {
        width: 100%;
        height: 100%;
        font-family: "微软雅黑";
    }
    ul,ol {
        list-style: none;
    }
    a {

        text-decoration: none;
    }
</style>
```

3.2.5.路由index.js文件

```
import Vue from 'vue'
import VueRouter from 'vue-router'
import Index from '../views/Index.vue'
import BusinessList from '../views/BusinessList.vue'
import BusinessInfo from '../views/BusinessInfo.vue'
import Login from '../views/Login.vue'
import Orders from '../views/Orders.vue'
import UserAddress from '../views/UserAddress.vue'
import Payment from '../views/Payment.vue'
import OrderList from '../views/OrderList.vue'
import AddUserAddress from '../views/AddUserAddress.vue'
import EditUserAddress from '../views/EditUserAddress.vue'
import Register from '../views/Register.vue'

Vue.use(VueRouter)

const routes = [{
        path: '/',
        name: 'Home',
        component: Index
    }, {
        path: '/index',
        name: 'Index',
        component: Index
    }, {
        path: '/businessList',
        name: 'BusinessList',
        component: BusinessList
    }, {
        path: '/businessInfo',
        name: 'BusinessInfo',
        component: BusinessInfo
    }, {
        path: '/login',
        name: 'Login',
        component: Login
    }, {
        path: '/orders',
        name: 'Orders',
        component: Orders
    }, {
        path: '/userAddress',
        name: 'UserAddress',
        component: UserAddress

    }, {
        path: '/payment',
        name: 'Payment',
        component: Payment
    }, {
        path: '/orderList',
        name: 'OrderList',
        component: OrderList
    }, {
        path: '/addUserAddress',
        name: 'AddUserAddress',
        component: AddUserAddress
    }, {
        path: '/editUserAddress',
        name: 'EditUserAddress',
        component: EditUserAddress
    }, {
        path: '/register',
        name: 'Register',
        component: Register
    }

]

//解决重复路由报异常问题
const originalPush = VueRouter.prototype.push;
VueRouter.prototype.push = function push(location) {
    return originalPush.call(this, location).catch(err => err)
}

const router = new VueRouter({
    mode: 'history',
    base: process.env.BASE_URL,
    routes
})

export default router
```

4.项目代码

4.1.领域模型（PO）

4.1.1.Business

```
package com.neusoft.elm.po;

public class Business {

    private Integer businessId;
    private String businessName;
    private String businessAddress;
    private String businessExplain;
    private String businessImg;
    private Integer orderTypeId;
    private double starPrice;            //起送费
    private double deliveryPrice;        //配送费
    private String remarks;

     //get、set ... ...
}
```

4.1.2.Cart

```
public class Cart {

    private Integer cartId;
    private Integer foodId;
    private Integer businessId;
    private String userId;
    private Integer quantity;

    //多对一：所属食品
    private Food food;
    //多对一：所属商家
    private Business business;

     //get、set ... ...
}
```

4.1.3.DeliveryAddress

```
public class DeliveryAddress {

    private Integer daId;
    private String contactName;
    private Integer contactSex;
    private String contactTel;
    private String address;
    private String userId;

     //get、set ... ...
}
```

4.1.4.Food

```
public class Food {

    private Integer foodId;
    private String foodName;
    private String foodExplain;
    private String foodImg;
    private Double foodPrice;
    private Integer businessId;
    private String remarks;

    //get、set ... ...
}
```

4.1.5.OrderDetailet

```
private Integer odId;

    private Integer odId;
    private Integer orderId;
    private Integer foodId;
    private Integer quantity;

    //多对一：所属食品
    private Food food;

    //get、set ... ...
}
```

4.1.6.Orders

```
public class Orders {

    private Integer orderId;
    private String userId;
    private Integer businessId;
    private String orderDate;
    private Double orderTotal;
    private Integer daId;          //送货地址编号
    private Integer orderState;    //订单状态（0：未支付；  1：已支付）

    //多对一：所属商家
    private Business business;
    //一对多：订单明细
    private List<OrderDetailet> list;

    //get、set ... ...

}
```

4.1.7.User

```
public class User {

    private String userId;
    private String password;
    private String userName;
    private Integer userSex;
    private String userImg;
    private Integer delTag;

    //get、set ... ...
}
```

4.2.服务器端代码

4.2.1.Mapper层代码

4.2.1.1.Business

```
package com.neusoft.elmboot.mapper;

import java.util.List;

import org.apache.ibatis.annotations.Mapper;
import org.apache.ibatis.annotations.Select;

import com.neusoft.elmboot.po.Business;

@Mapper
public interface BusinessMapper {

    @Select("select * from business where orderTypeId=#{orderTypeId} order by businessId")
    public List<Business> listBusinessByOrderTypeId(Integer orderTypeId);

    @Select("select * from business where businessId=#{businessId}")
    public Business getBusinessById(Integer businessId);
}
```

4.2.1.2.Cart

```
package com.neusoft.elmboot.mapper;

import java.util.List;

import org.apache.ibatis.annotations.Insert;
import org.apache.ibatis.annotations.Mapper;
import org.apache.ibatis.annotations.Update;

import com.neusoft.elmboot.po.Cart;

@Mapper
public interface CartMapper {

    public List<Cart> listCart(Cart cart);

    @Insert("insert into cart values(null,#{foodId},#{businessId},#{userId},1)")
    public int saveCart(Cart cart);

    @Update("update cart set quantity=#{quantity} where foodId=#{foodId} and businessId=#
{businessId} and userId=#{userId}")
    public int updateCart(Cart cart);

    public int removeCart(Cart cart);
}
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
    "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
<mapper namespace="com.neusoft.elmboot.mapper.CartMapper">

    <resultMap type="Cart" id="cartResultMap">
        <id column="cartId" property="cartId"/>
        <result column="foodId" property="foodId"/>
        <result column="businessId" property="businessId"/>
        <result column="userId" property="userId"/>
        <result column="quantity" property="quantity"/>
        <association property="food" javaType="Food"
             select="com.neusoft.elmboot.mapper.FoodMapper.getFoodById" column="foodId"/>
        <association property="business" javaType="Business"
             select="com.neusoft.elmboot.mapper.BusinessMapper.getBusinessById"
column="businessId"/>

    </resultMap>

    <select id="listCart" parameterType="Cart" resultMap="cartResultMap">
        select * from cart
        <where>
            userId=#{userId}
            <if test="businessId!=null and businessId!=''">
                and businessId=#{businessId}
            </if>
        </where>
        order by cartId
    </select>

    <delete id="removeCart" parameterType="Cart">
        delete from cart
        <where>
            userId=#{userId} and businessId=#{businessId}
            <if test="foodId!=null and foodId!=''">
                and foodId=#{foodId}
            </if>
        </where>
    </delete>

</mapper>
```

4.2.1.3.DeliveryAddress

```
package com.neusoft.elmboot.mapper;

import java.util.List;

import org.apache.ibatis.annotations.Delete;
import org.apache.ibatis.annotations.Insert;
import org.apache.ibatis.annotations.Mapper;
import org.apache.ibatis.annotations.Select;
import org.apache.ibatis.annotations.Update;

import com.neusoft.elmboot.po.DeliveryAddress;

@Mapper
public interface DeliveryAddressMapper {

    @Select("select * from deliveryAddress where userId=#{userId} order by daId")
    public List<DeliveryAddress> listDeliveryAddressByUserId(String userId);

    @Select("select * from deliveryAddress where daId=#{daId}")
    public DeliveryAddress getDeliveryAddressById(Integer daId);

    @Insert("insert into deliveryAddress values(null,#{contactName},#{contactSex},#{contactTel},#{address},#{userId})")
    public int saveDeliveryAddress(DeliveryAddress deliveryAddress);

    @Update("update deliveryAddress set contactName=#{contactName},contactSex=#{contactSex},contactTel=#{contactTel},address=#{address} where daId=#{daId}")
    public int updateDeliveryAddress(DeliveryAddress deliveryAddress);

    @Delete("delete from deliveryAddress where daId=#{daId}")

    public int removeDeliveryAddress(Integer daId);
}
```

4.2.1.4.Food

```
package com.neusoft.elmboot.mapper;

import java.util.List;

import org.apache.ibatis.annotations.Mapper;
import org.apache.ibatis.annotations.Select;

import com.neusoft.elmboot.po.Food;

@Mapper
public interface FoodMapper {

    @Select("select * from food where businessId=#{businessId} order by foodId")
    public List<Food> listFoodByBusinessId(Integer businessId);

    @Select("select * from food where foodId=#{foodId}")
    public Food getFoodById(Integer foodId);
}
```

4.2.1.5.OrderDetailet

```
package com.neusoft.elmboot.mapper;

import java.util.List;
import org.apache.ibatis.annotations.Mapper;

import com.neusoft.elmboot.po.OrderDetailet;

@Mapper
public interface OrderDetailetMapper {

    public int saveOrderDetailetBatch(List<OrderDetailet> list);

    public List<OrderDetailet> listOrderDetailetByOrderId(Integer orderOd);
}
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
    "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
<mapper namespace="com.neusoft.elmboot.mapper.OrderDetailetMapper">


    <insert id="saveOrderDetailetBatch" parameterType="OrderDetailet">
```

```
        insert into orderDetailet(orderId,foodId,quantity) values
        <foreach collection="list" item="od" separator=",">
            (#{od.orderId},#{od.foodId},#{od.quantity})
        </foreach>
    </insert>

    <resultMap type="OrderDetailet" id="orderDetailetResultMap">
        <id column="odId" property="odId"/>
        <result column="orderId" property="orderId"/>
        <result column="foodId" property="foodId"/>
        <result column="quantity" property="quantity"/>
        <association property="food" javaType="Food"
             select="com.neusoft.elmboot.mapper.FoodMapper.getFoodById" column="foodId"/>
    </resultMap>

    <select id="listOrderDetailetByOrderId" parameterType="Integer"
resultMap="orderDetailetResultMap">
        select * from orderDetailet where orderId=#{orderId}
    </select>

</mapper>
```

4.2.1.6.Orders

```
package com.neusoft.elmboot.mapper;

import org.apache.ibatis.annotations.Insert;
import org.apache.ibatis.annotations.Mapper;
import org.apache.ibatis.annotations.Select;

import com.neusoft.elmboot.po.User;

@Mapper
public interface UserMapper {

    @Select("select * from user where userId=#{userId} and password=#{password}")
    public User getUserByIdByPass(User user);

    @Select("select count(*) from user where userId=#{userId}")
    public int getUserById(String userId);

    @Insert("insert into user values(#{userId},#{password},#{userName},#{userSex},null,1)")
    public int saveUser(User user);
}
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
    "http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<mapper namespace="com.neusoft.elmboot.mapper.OrdersMapper">
```

```
    <resultMap type="Orders" id="ordersResultMap">
        <id column="orderId" property="orderId"/>
        <result column="userId" property="userId"/>
        <result column="businessId" property="businessId"/>
        <result column="orderDate" property="orderDate"/>
        <result column="orderTotal" property="orderTotal"/>
        <result column="daId" property="daId"/>
        <result column="orderState" property="orderState"/>
        <association property="business" javaType="Business"
             select="com.neusoft.elmboot.mapper.BusinessMapper.getBusinessById"
column="businessId"/>
        <collection property="list" ofType="OrderDetailet"
             select="com.neusoft.elmboot.mapper.OrderDetailetMapper.listOrderDetailetByOrderId"
column="orderId"/>
    </resultMap>

    <select id="getOrdersById" parameterType="Integer" resultMap="ordersResultMap">
        select * from orders where orderId=#{orderId}
    </select>

    <select id="listOrdersByUserId" parameterType="String" resultMap="ordersResultMap">
        select * from orders where userId=#{userId} order by orderId
    </select>

</mapper>
```

4.2.1.7.User

```
package com.neusoft.elmboot.mapper;

import org.apache.ibatis.annotations.Insert;
import org.apache.ibatis.annotations.Mapper;
import org.apache.ibatis.annotations.Select;

import com.neusoft.elmboot.po.User;

@Mapper
public interface UserMapper {

    @Select("select * from user where userId=#{userId} and password=#{password}")
    public User getUserByIdByPass(User user);

    @Select("select count(*) from user where userId=#{userId}")
    public int getUserById(String userId);

    @Insert("insert into user values(#{userId},#{password},#{userName},#{userSex},null,1)")
    public int saveUser(User user);
}
```

4.2.2.Service层代码

4.2.2.1.Business

```
package com.neusoft.elmboot.service;

import java.util.List;

import com.neusoft.elmboot.po.Business;

public interface BusinessService {

    public List<Business> listBusinessByOrderTypeId(Integer orderTypeId);
    public Business getBusinessById(Integer businessId);
}
```

```
package com.neusoft.elmboot.service.impl;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import com.neusoft.elmboot.mapper.BusinessMapper;
import com.neusoft.elmboot.po.Business;
import com.neusoft.elmboot.service.BusinessService;

@Service
public class BusinessServiceImpl implements BusinessService{

    @Autowired
    private BusinessMapper businessMapper;

    @Override
    public List<Business> listBusinessByOrderTypeId(Integer orderTypeId) {
        return businessMapper.listBusinessByOrderTypeId(orderTypeId);
    }

    @Override
    public Business getBusinessById(Integer businessId) {
        return businessMapper.getBusinessById(businessId);
    }
}
```

4.2.2.2.Cart

```java
package com.neusoft.elmboot.service;

import java.util.List;

import com.neusoft.elmboot.po.Cart;

public interface CartService {

    public List<Cart> listCart(Cart cart);
    public int saveCart(Cart cart);
    public int updateCart(Cart cart);
    public int removeCart(Cart cart);
}
```

```java
package com.neusoft.elmboot.service.impl;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import com.neusoft.elmboot.mapper.CartMapper;
import com.neusoft.elmboot.po.Cart;
import com.neusoft.elmboot.service.CartService;

@Service
public class CartServiceImpl implements CartService{

    @Autowired
    private CartMapper cartMapper;

    @Override
    public List<Cart> listCart(Cart cart) {
        return cartMapper.listCart(cart);
    }

    @Override
    public int saveCart(Cart cart) {
        return cartMapper.saveCart(cart);
    }

    @Override
    public int updateCart(Cart cart) {
        return cartMapper.updateCart(cart);
    }

    @Override
    public int removeCart(Cart cart) {
        return cartMapper.removeCart(cart);
    }
}
```

4.2.2.3.DeliveryAddress

```
package com.neusoft.elmboot.service;

import java.util.List;

import com.neusoft.elmboot.po.DeliveryAddress;

public interface DeliveryAddressService {

    public List<DeliveryAddress> listDeliveryAddressByUserId(String userId);

    public DeliveryAddress getDeliveryAddressById(Integer daId);
    public int saveDeliveryAddress(DeliveryAddress deliveryAddress);
    public int updateDeliveryAddress(DeliveryAddress deliveryAddress);
    public int removeDeliveryAddress(Integer daId);
}
```

```
package com.neusoft.elmboot.service.impl;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import com.neusoft.elmboot.mapper.DeliveryAddressMapper;
import com.neusoft.elmboot.po.DeliveryAddress;
import com.neusoft.elmboot.service.DeliveryAddressService;

@Service
public class DeliveryAddressServiceImpl implements DeliveryAddressService{

    @Autowired
    private DeliveryAddressMapper deliveryAddressMapper;

    @Override
    public List<DeliveryAddress> listDeliveryAddressByUserId(String userId) {
        return deliveryAddressMapper.listDeliveryAddressByUserId(userId);
    }

    @Override
    public DeliveryAddress getDeliveryAddressById(Integer daId) {
        return deliveryAddressMapper.getDeliveryAddressById(daId);
    }

    @Override
    public int saveDeliveryAddress(DeliveryAddress deliveryAddress) {
        return deliveryAddressMapper.saveDeliveryAddress(deliveryAddress);
    }
    @Override
    public int updateDeliveryAddress(DeliveryAddress deliveryAddress) {
        return deliveryAddressMapper.updateDeliveryAddress(deliveryAddress);
    }

    @Override
    public int removeDeliveryAddress(Integer daId) {
        return deliveryAddressMapper.removeDeliveryAddress(daId);
    }
}
```

4.2.2.4.Food

```
package com.neusoft.elmboot.service;

import java.util.List;

import com.neusoft.elmboot.po.Food;

public interface FoodService {

    public List<Food> listFoodByBusinessId(Integer businessId);
}
```

```
package com.neusoft.elmboot.service.impl;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import com.neusoft.elmboot.mapper.FoodMapper;
import com.neusoft.elmboot.po.Food;
import com.neusoft.elmboot.service.FoodService;

@Service
public class FoodServiceImpl implements FoodService{

    @Autowired
    private FoodMapper foodMapper;

    @Override
    public List<Food> listFoodByBusinessId(Integer businessId) {
        return foodMapper.listFoodByBusinessId(businessId);
    }
}
```

4.2.2.5.Orders

```java
package com.neusoft.elmboot.service;

import java.util.List;

import com.neusoft.elmboot.po.Orders;

public interface OrdersService {

    public int createOrders(Orders orders);
    public Orders getOrdersById(Integer orderId);
    public List<Orders> listOrdersByUserId(String userId);
}
```

```java
package com.neusoft.elmboot.service.impl;

import java.util.ArrayList;
import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import org.springframework.transaction.annotation.Transactional;

import com.neusoft.elmboot.mapper.CartMapper;
import com.neusoft.elmboot.mapper.OrderDetailetMapper;
import com.neusoft.elmboot.mapper.OrdersMapper;
import com.neusoft.elmboot.po.Cart;
import com.neusoft.elmboot.po.OrderDetailet;
import com.neusoft.elmboot.po.Orders;
import com.neusoft.elmboot.service.OrdersService;
import com.neusoft.elmboot.util.CommonUtil;

@Service
public class OrdersServiceImpl implements OrdersService{

    @Autowired
    private CartMapper cartMapper;
    @Autowired
    private OrdersMapper ordersMapper;
    @Autowired
    private OrderDetailetMapper orderDetailetMapper;

    @Override
    @Transactional
    public int createOrders(Orders orders) {
        //1、查询当前用户购物车中当前商家的所有食品
        Cart cart = new Cart();
        cart.setUserId(orders.getUserId());
        cart.setBusinessId(orders.getBusinessId());
        List<Cart> cartList = cartMapper.listCart(cart);
        //2、创建订单（返回生成的订单编号）
        orders.setOrderDate(CommonUtil.getCurrentDate());
        ordersMapper.saveOrders(orders);
        int orderId = orders.getOrderId();

        //3、批量添加订单明细
        List<OrderDetailet> list = new ArrayList<>();
        for(Cart c : cartList) {
            OrderDetailet od = new OrderDetailet();
            od.setOrderId(orderId);
            od.setFoodId(c.getFoodId());
            od.setQuantity(c.getQuantity());
            list.add(od);
        }
        orderDetailetMapper.saveOrderDetailetBatch(list);

        //4、从购物车表中删除相关食品信息
        cartMapper.removeCart(cart);

        return orderId;
    }

    @Override
    public Orders getOrdersById(Integer orderId) {
        return ordersMapper.getOrdersById(orderId);
    }

    @Override
    public List<Orders> listOrdersByUserId(String userId){
        return ordersMapper.listOrdersByUserId(userId);
    }
}
```

4.2.2.6.user

```
package com.neusoft.elmboot.service;

import com.neusoft.elmboot.po.User;

public interface UserService {

    public User getUserByIdByPass(User user);
    public int getUserById(String userId);
    public int saveUser(User user);
}
```

```
package com.neusoft.elmboot.service.impl;


import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import com.neusoft.elmboot.mapper.UserMapper;
import com.neusoft.elmboot.po.User;
import com.neusoft.elmboot.service.UserService;

@Service
public class UserServiceImpl implements UserService{

    @Autowired
    private UserMapper userMapper;

    @Override
    public User getUserByIdByPass(User user) {
        return userMapper.getUserByIdByPass(user);
    }

    @Override
    public int getUserById(String userId) {
        return userMapper.getUserById(userId);
    }

    @Override
    public int saveUser(User user) {
        return userMapper.saveUser(user);
    }
}
```

4.2.3.Controller层代码

4.2.3.1.Business

```
package com.neusoft.elmboot.controller;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import com.neusoft.elmboot.po.Business;
import com.neusoft.elmboot.service.BusinessService;

@RestController
@RequestMapping("/BusinessController")
public class BusinessController {


    @Autowired
    private BusinessService businessService;

    @RequestMapping("/listBusinessByOrderTypeId")
    public List<Business> listBusinessByOrderTypeId(Business business) throws Exception{
        return businessService.listBusinessByOrderTypeId(business.getOrderTypeId());
    }

    @RequestMapping("/getBusinessById")
    public Business getBusinessById(Business business) throws Exception{
        return businessService.getBusinessById(business.getBusinessId());
    }

}
```

4.2.3.2.Cart

```
package com.neusoft.elmboot.controller;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import com.neusoft.elmboot.po.Cart;
import com.neusoft.elmboot.service.CartService;

@RestController
@RequestMapping("/CartController")
public class CartController {

    @Autowired
    private CartService cartService;

    @RequestMapping("/listCart")
    public List<Cart> listCart(Cart cart) throws Exception{
        return cartService.listCart(cart);
    }

    @RequestMapping("/saveCart")
    public int saveCart(Cart cart) throws Exception{
        return cartService.saveCart(cart);
    }

    @RequestMapping("/updateCart")
    public int updateCart(Cart cart) throws Exception{
        return cartService.updateCart(cart);
    }


    @RequestMapping("/removeCart")
    public int removeCart(Cart cart) throws Exception{
        return cartService.removeCart(cart);
    }
}
```

4.2.3.3.DeliveryAddress

```
package com.neusoft.elmboot.controller;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import com.neusoft.elmboot.po.DeliveryAddress;
import com.neusoft.elmboot.service.DeliveryAddressService;

@RestController
@RequestMapping("/DeliveryAddressController")
public class DeliveryAddressController {

    @Autowired
    private DeliveryAddressService deliveryAddressService;

    @RequestMapping("/listDeliveryAddressByUserId")
    public List<DeliveryAddress> listDeliveryAddressByUserId(DeliveryAddress deliveryAddress)
throws Exception{
        return deliveryAddressService.listDeliveryAddressByUserId(deliveryAddress.getUserId());
    }

    @RequestMapping("/getDeliveryAddressById")
    public DeliveryAddress getDeliveryAddressById(DeliveryAddress deliveryAddress) throws
Exception{
        return deliveryAddressService.getDeliveryAddressById(deliveryAddress.getDaId());
    }

    @RequestMapping("/saveDeliveryAddress")
    public int saveDeliveryAddress(DeliveryAddress deliveryAddress) throws Exception{
        return deliveryAddressService.saveDeliveryAddress(deliveryAddress);
    }

    @RequestMapping("/updateDeliveryAddress")
    public int updateDeliveryAddress(DeliveryAddress deliveryAddress) throws Exception{
        return deliveryAddressService.updateDeliveryAddress(deliveryAddress);
    }

    @RequestMapping("/removeDeliveryAddress")
    public int removeDeliveryAddress(DeliveryAddress deliveryAddress) throws Exception{

        return deliveryAddressService.removeDeliveryAddress(deliveryAddress.getDaId());
    }
}
```

4.2.3.4.Food

```
package com.neusoft.elmboot.controller;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import com.neusoft.elmboot.po.Food;
import com.neusoft.elmboot.service.FoodService;

@RestController
@RequestMapping("/FoodController")
public class FoodController {

    @Autowired
    private FoodService foodService;

    @RequestMapping("/listFoodByBusinessId")
    public List<Food> listFoodByBusinessId(Food food) throws Exception{
        return foodService.listFoodByBusinessId(food.getBusinessId());
    }
}
```

4.2.3.5.Orders

```
package com.neusoft.elmboot.controller;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import com.neusoft.elmboot.po.Orders;
import com.neusoft.elmboot.service.OrdersService;

@RestController
@RequestMapping("/OrdersController")
public class OrdersController {


    @Autowired
    private OrdersService ordersService;

    @RequestMapping("/createOrders")
    public int createOrders(Orders orders) throws Exception{
        return ordersService.createOrders(orders);
    }

    @RequestMapping("/getOrdersById")
    public Orders getOrdersById(Orders orders) throws Exception{
        return ordersService.getOrdersById(orders.getOrderId());
    }

    @RequestMapping("/listOrdersByUserId")
    public List<Orders> listOrdersByUserId(Orders orders) throws Exception{
        return ordersService.listOrdersByUserId(orders.getUserId());
    }
}
```

4.2.3.7.user

```
package com.neusoft.elmboot.controller;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import com.neusoft.elmboot.po.User;
import com.neusoft.elmboot.service.UserService;

@RestController
@RequestMapping("/UserController")
public class UserController {

    @Autowired
    private UserService userService;

    @RequestMapping("/getUserByIdByPass")
    public User getUserByIdByPass(User user) throws Exception{
        return userService.getUserByIdByPass(user);
    }

    @RequestMapping("/getUserById")
    public int getUserById(User user) throws Exception{
        return userService.getUserById(user.getUserId());
    }

    @RequestMapping("/saveUser")
    public int saveUser(User user) throws Exception{
        return userService.saveUser(user);

    }
}
```

4.3.前端代码

4.3.1.Index组件

> 注意点:
>
> 1. 修改图片路径，包括css中**背景图片路径**。
> 2. style中添加scoped。
> 3. 将静态工程中的 icon.css内容添加到style中。

```
<template>
    <div class="wrapper">
        <!-- header部分 -->
        <header>
            <div class="icon-location-box">
                <div class="icon-location"></div>
            </div>
            <div class="location-text">沈阳市规划大厦<i class="fa fa-caret-down"></i></div>
        </header>

        <!-- search部分 -->
        <!--
                    搜索框部分（此块与search-fixed-top块宽度高度一致，用于当
                    search-fixed-top块固定后，挡住下面块不要窜上去）
                -->
        <div class="search">
            <!-- 当滚动条超过上面的定位块时，search-fixed-top块变成固定在顶部。 -->
            <div class="search-fixed-top" ref="fixedBox">
                <!-- 搜索框部分中间的白框 -->
                <div class="search-box">
                    <i class="fa fa-search"></i>搜索饿了么商家、商品名称
                </div>
            </div>
        </div>

        <!-- 点餐分类部分 -->
        <ul class="foodtype">
            <li @click="toBusinessList(1)">
                <img src="../assets/dcfl01.png">
                <p>美食</p>
            </li>
            <li @click="toBusinessList(2)">
                <img src="../assets/dcfl02.png">
                <p>早餐</p>
            </li>

            <li @click="toBusinessList(3)">
                <img src="../assets/dcfl03.png">
                <p>跑腿代购</p>
            </li>
            <li @click="toBusinessList(4)">
                <img src="../assets/dcfl04.png">
                <p>汉堡披萨</p>
            </li>
            <li @click="toBusinessList(5)">
                <img src="../assets/dcfl05.png">
                <p>甜品饮品</p>
            </li>
            <li @click="toBusinessList(6)">
                <img src="../assets/dcfl06.png">
                <p>速食简餐</p>
            </li>
            <li @click="toBusinessList(7)">
                <img src="../assets/dcfl07.png">
                <p>地方小吃</p>
            </li>
            <li @click="toBusinessList(8)">
                <img src="../assets/dcfl08.png">
                <p>米粉面馆</p>
            </li>
            <li @click="toBusinessList(9)">
                <img src="../assets/dcfl09.png">
                <p>包子粥铺</p>
            </li>
            <li @click="toBusinessList(10)">
                <img src="../assets/dcfl10.png">
                <p>炸鸡炸串</p>
            </li>
        </ul>

        <!-- 横幅广告部分（注意：此处有背景图片） -->
        <div class="banner">
            <h3>品质套餐</h3>
            <p>搭配齐全吃得好</p>
            <a>立即抢购 &gt;</a>
        </div>

        <!-- 超级会员部分 -->
        <div class="supermember">
            <div class="left">
                <img src="../assets/super_member.png">
                <h3>超级会员</h3>
                <p>&#8226; 每月享超值权益</p>
            </div>
            <div class="right">
                立即开通 &gt;
            </div>
        </div>


        <!-- 推荐商家部分 -->
```

```
        <div class="recommend">
            <div class="recommend-line"></div>
            <p>推荐商家</p>
            <div class="recommend-line"></div>
        </div>

        <!-- 推荐方式部分 -->
        <ul class="recommendtype">
            <li>综合排序<i class="fa fa-caret-down"></i></li>
            <li>距离最近</li>
            <li>销量最高</li>
            <li>筛选<i class="fa fa-filter"></i></li>
        </ul>

        <!-- 推荐商家列表部分 -->
        <ul class="business">
            <li>
                <img src="../assets/sj01.png">
                <div class="business-info">
                    <div class="business-info-h">
                        <h3>万家饺子（软件园E18店）</h3>
                        <div class="business-info-like">&#8226;</div>
                    </div>
                    <div class="business-info-star">
                        <div class="business-info-star-left">
                            <i class="fa fa-star"></i>
                            <i class="fa fa-star"></i>
                            <i class="fa fa-star"></i>
                            <i class="fa fa-star"></i>
                            <i class="fa fa-star"></i>
                            <p>4.9 月售345单</p>
                        </div>
                        <div class="business-info-star-right">
                            蜂鸟专送
                        </div>
                    </div>
                    <div class="business-info-delivery">
                        <p>&#165;15起送 | &#165;3配送</p>
                        <p>3.22km | 30分钟</p>
                    </div>
                    <div class="business-info-explain">
                        <div>各种饺子</div>
                    </div>
                    <div class="business-info-promotion">
                        <div class="business-info-promotion-left">
                            <div class="business-info-promotion-left-incon">新</div>
                            <p>饿了么新用户首单立减9元</p>
                        </div>
                        <div class="business-info-promotion-right">
                            <p>2个活动</p>
                            <i class="fa fa-caret-down"></i>
                        </div>

                    </div>
                    <div class="business-info-promotion">
                        <div class="business-info-promotion-left">
                            <div class="business-info-promotion-left-incon" style="background-color: #F1884F;">特</div>
                            <p>特价商品5元起</p>
                        </div>
                    </div>
                </div>
            </li>
            <li>
                <img src="../assets/sj02.png">
                <div class="business-info">
                    <div class="business-info-h">
                        <h3>小锅饭豆腐馆（全运店）</h3>
                        <div class="business-info-like">&#8226;</div>
                    </div>
                    <div class="business-info-star">
                        <div class="business-info-star-left">
                            <i class="fa fa-star"></i>
                            <i class="fa fa-star"></i>
                            <i class="fa fa-star"></i>
                            <i class="fa fa-star"></i>
                            <i class="fa fa-star"></i>
                            <p>4.9 月售345单</p>
                        </div>
                        <div class="business-info-star-right">
                            蜂鸟专送
                        </div>
                    </div>
                    <div class="business-info-delivery">
                        <p>&#165;15起送 | &#165;3配送</p>
                        <p>3.22km | 30分钟</p>
                    </div>
                    <div class="business-info-explain">
                        <div>各种饺子</div>
                    </div>
                    <div class="business-info-promotion">
                        <div class="business-info-promotion-left">
                            <div class="business-info-promotion-left-incon">新</div>
                            <p>饿了么新用户首单立减9元</p>
                        </div>
                        <div class="business-info-promotion-right">
                            <p>2个活动</p>
                            <i class="fa fa-caret-down"></i>
                        </div>
                    </div>
                    <div class="business-info-promotion">
                        <div class="business-info-promotion-left">
                            <div class="business-info-promotion-left-incon">特</div>
                            <p>特价商品5元起</p>
                        </div>
                    </div>

                </div>
            </li>
            <li>
                <img src="../assets/sj03.png">
                <div class="business-info">
                    <div class="business-info-h">
                        <h3>麦当劳麦乐送（全运路店）</h3>
                        <div class="business-info-like">&#8226;</div>
                    </div>
                    <div class="business-info-star">
                        <div class="business-info-star-left">
                            <i class="fa fa-star"></i>
                            <i class="fa fa-star"></i>
                            <i class="fa fa-star"></i>
                            <i class="fa fa-star"></i>
                            <i class="fa fa-star"></i>
                            <p>4.9 月售345单</p>
                        </div>
                        <div class="business-info-star-right">
                            蜂鸟专送
                        </div>
                    </div>
                    <div class="business-info-delivery">
                        <p>&#165;15起送 | &#165;3配送</p>
                        <p>3.22km | 30分钟</p>
                    </div>
                    <div class="business-info-explain">
                        <div>各种饺子</div>
                    </div>
                    <div class="business-info-promotion">
                        <div class="business-info-promotion-left">
                            <div class="business-info-promotion-left-incon">新</div>
                            <p>饿了么新用户首单立减9元</p>
                        </div>
                        <div class="business-info-promotion-right">
                            <p>2个活动</p>
                            <i class="fa fa-caret-down"></i>
                        </div>
                    </div>
                    <div class="business-info-promotion">
                        <div class="business-info-promotion-left">
                            <div class="business-info-promotion-left-incon">特</div>
                            <p>特价商品5元起</p>
                        </div>
                    </div>
                </div>
            </li>
            <li>
                <img src="../assets/sj04.png">
                <div class="business-info">
                    <div class="business-info-h">
                        <h3>米村拌饭（浑南店）</h3>
                        <div class="business-info-like">&#8226;</div>

                    </div>
                    <div class="business-info-star">
                        <div class="business-info-star-left">
                            <i class="fa fa-star"></i>
                            <i class="fa fa-star"></i>
                            <i class="fa fa-star"></i>
                            <i class="fa fa-star"></i>
                            <i class="fa fa-star"></i>
                            <p>4.9 月售345单</p>
                        </div>
                        <div class="business-info-star-right">
                            蜂鸟专送
                        </div>
                    </div>
                    <div class="business-info-delivery">
                        <p>&#165;15起送 | &#165;3配送</p>
                        <p>3.22km | 30分钟</p>
                    </div>
                    <div class="business-info-explain">
                        <div>各种饺子</div>
                    </div>
                    <div class="business-info-promotion">
                        <div class="business-info-promotion-left">
                            <div class="business-info-promotion-left-incon">新</div>
                            <p>饿了么新用户首单立减9元</p>
                        </div>
                        <div class="business-info-promotion-right">
                            <p>2个活动</p>
                            <i class="fa fa-caret-down"></i>
                        </div>
                    </div>
                    <div class="business-info-promotion">
                        <div class="business-info-promotion-left">
                            <div class="business-info-promotion-left-incon">特</div>
                            <p>特价商品5元起</p>
                        </div>
                    </div>
                </div>
            </li>
            <li>
                <img src="../assets/sj05.png">
                <div class="business-info">
                    <div class="business-info-h">
                        <h3>串记串道（中海康城店）</h3>
                        <div class="business-info-like">&#8226;</div>
                    </div>
                    <div class="business-info-star">
                        <div class="business-info-star-left">
                            <i class="fa fa-star"></i>
                            <i class="fa fa-star"></i>
                            <i class="fa fa-star"></i>
                            <i class="fa fa-star"></i>
                            <i class="fa fa-star"></i>

                            <p>4.9 月售345单</p>
                        </div>
                        <div class="business-info-star-right">
                            蜂鸟专送
                        </div>
                    </div>
                    <div class="business-info-delivery">
                        <p>&#165;15起送 | &#165;3配送</p>
                        <p>3.22km | 30分钟</p>
                    </div>
                    <div class="business-info-explain">
                        <div>各种饺子</div>
                    </div>
                    <div class="business-info-promotion">
                        <div class="business-info-promotion-left">
                            <div class="business-info-promotion-left-incon">新</div>
                            <p>饿了么新用户首单立减9元</p>
                        </div>
                        <div class="business-info-promotion-right">
                            <p>2个活动</p>
                            <i class="fa fa-caret-down"></i>
                        </div>
                    </div>
                    <div class="business-info-promotion">
                        <div class="business-info-promotion-left">
                            <div class="business-info-promotion-left-incon">特</div>
                            <p>特价商品5元起</p>
                        </div>
                    </div>
                </div>
            </li>
        </ul>

        <!-- 底部菜单部分 -->
        <Footer></Footer>
    </div>
</template>

<script>
    //导入共通组件
    import Footer from '../components/Footer.vue';

    export default {
        name: 'Index',
        mounted() {
            document.onscroll = ()=> {
                //获取滚动条位置
                let s1 = document.documentElement.scrollTop;
                let s2 = document.body.scrollTop;
                let scroll = s1 == 0 ? s2 : s1;
                //获取视口宽度
                let width = document.documentElement.clientWidth;


                //获取顶部固定块
                let search = this.$refs.fixedBox;

                //判断滚动条超过视口宽度的12%时，搜索块变固定定位
                if (scroll > width * 0.12) {
                    search.style.position = 'fixed';
                    search.style.left = '0';
                    search.style.top = '0';
                } else {
                    search.style.position = 'static';
                }
            }
        },
        destroyed() {
            //当切换到其他组件时，就不需要document滚动条事件，所以将此事件去掉
            document.onscroll = null;
        },
        components:{
            Footer
        },
        methods:{
            toBusinessList(orderTypeId){
                this.$router.push({path:'/businessList',query:{orderTypeId:orderTypeId}});
            }
        }
    }
</script>

<style scoped>
    /****************** 总容器 ******************/
    .wrapper {
        width: 100%;
        height: 100%;
    }

    /****************** header ******************/
    .wrapper header {
        width: 100%;
        height: 12vw;
        background-color: #0097FF;

        display: flex;
        align-items: center;
    }

    .wrapper header .icon-location-box {
        width: 3.5vw;
        height: 3.5vw;
        margin: 0 1vw 0 3vw;
    }

    .wrapper header .location-text {
        font-size: 4.5vw;

        font-weight: 700;
        color: #fff;
    }

    .wrapper header .location-text .fa-caret-down {
        margin-left: 1vw;
    }

    /****************** search ******************/
    .wrapper .search {
        width: 100%;
        height: 13vw;
    }

    .wrapper .search .search-fixed-top {
        width: 100%;
        height: 13vw;
        background-color: #0097FF;
        display: flex;
        justify-content: center;
        align-items: center;
    }

    .wrapper .search .search-fixed-top .search-box {
        width: 90%;
        height: 9vw;
        background-color: #fff;
        border-radius: 2px;

        display: flex;
        justify-content: center;
        align-items: center;

        font-size: 3.5vw;
        color: #AEAEAE;
        font-family: "宋体";
        /*此样式是让文本选中状态无效*/
        user-select: none;
    }

    .wrapper .search .search-fixed-top .search-box .fa-search {
        margin-right: 1vw;
    }

    /****************** 点餐分类部分 ******************/
    .wrapper .foodtype {
        width: 100%;
        height: 48vw;

        display: flex;
        flex-wrap: wrap;
        justify-content: space-around;
        /*要使用align-content。10个子元素将自动换行为两行，而且两行作为一个整体垂直居中*/

        align-content: center;
    }

    .wrapper .foodtype li {
        /*一共10个子元素，通过计算，子元素宽度在16.7 ~ 20 之间，才能保证换两行*/
        width: 18vw;
        height: 20vw;

        display: flex;
        /*弹性盒子主轴方向设为column，然后仍然是垂直水平方向居中*/
        flex-direction: column;
        justify-content: center;
        align-items: center;

        user-select: none;
        cursor: pointer;
    }

    .wrapper .foodtype li img {
        width: 12vw;
        /*视频讲解时高度设置为12vw，实际上设置为10.3vw更佳*/
        height: 10.3vw;
    }

    .wrapper .foodtype li p {
        font-size: 3.2vw;
        color: #666;
    }

    /****************** 横幅广告部分 ******************/
    .wrapper .banner {
        /**
         * 设置容器宽度95%，然后水平居中，这样两边留白；
         * 这里不能用padding，因为背景图片也会覆盖padding
         */
        width: 95%;
        margin: 0 auto;
        height: 29vw;

        /*此三个样式组合，可以保证背景图片充满整个容器*/
        background-image: url(../assets/index_banner.png);
        background-repeat: no-repeat;
        background-size: cover;

        box-sizing: border-box;
        padding: 2vw 6vw;
    }

    .wrapper .banner h3 {
        font-size: 4.2vw;
        margin-bottom: 1.2vw;
    }


    .wrapper .banner p {
        font-size: 3.4vw;
        color: #666;
        margin-bottom: 2.4vw;
    }

    .wrapper .banner a {
        font-size: 3vw;
        color: #C79060;
        font-weight: 700;
    }

    /****************** 超级会员部分 ******************/
    .wrapper .supermember {
        /*这里也设置容器宽度95%，不能用padding，因为背景色也会充满padding*/
        width: 95%;
        margin: 0 auto;
        height: 11.5vw;
        background-color: #FEEDC1;
        margin-top: 1.3vw;
        border-radius: 2px;
        color: #644F1B;

        display: flex;
        justify-content: space-between;
        align-items: center;
    }

    .wrapper .supermember .left {
        display: flex;
        align-items: center;
        margin-left: 4vw;
        user-select: none;
    }

    .wrapper .supermember .left img {
        width: 6vw;
        height: 6vw;
        margin-right: 2vw;
    }

    .wrapper .supermember .left h3 {
        font-size: 4vw;
        margin-right: 2vw;
    }

    .wrapper .supermember .left p {
        font-size: 3vw;
    }

    .wrapper .supermember .right {
        font-size: 3vw;
        margin-right: 4vw;

        cursor: pointer;
```

```
}

/****************** 推荐商家部分 ******************/
.wrapper .recommend {
    width: 100%;
    height: 14vw;
    display: flex;
    justify-content: center;
    align-items: center;
}

.wrapper .recommend .recommend-line {
    width: 6vw;
    height: 0.2vw;
    background-color: #888;
}

.wrapper .recommend p {
    font-size: 4vw;
    margin: 0 4vw;
}

/****************** 推荐方式部分 ******************/
.wrapper .recommendtype {
    width: 100%;
    height: 5vw;
    margin-bottom: 5vw;

    display: flex;
    justify-content: space-around;
    align-items: center;
}

.wrapper .recommendtype li {
    font-size: 3.5vw;
    color: #555;
}

/****************** 推荐商家列表部分 ******************/
.wrapper .business {
    width: 100%;
    margin-bottom: 14vw;
}

.wrapper .business li {
    width: 100%;
    box-sizing: border-box;
    padding: 2.5vw;
    user-select: none;
    border-bottom: solid 1px #DDD;

    display: flex;

}
```

```
    .wrapper .business li img {
        width: 18vw;
        height: 18vw;
    }

    .wrapper .business li .business-info {
        width: 100%;
        box-sizing: border-box;
        padding-left: 3vw;
    }

    .wrapper .business li .business-info .business-info-h {
        display: flex;
        justify-content: space-between;
        align-items: center;
        margin-bottom: 2vw;
    }

    .wrapper .business li .business-info .business-info-h h3 {
        font-size: 4vw;
        color: #333;
    }

    .wrapper .business li .business-info .business-info-h .business-info-like {
        width: 1.6vw;
        height: 3.4vw;
        background-color: #666;
        color: #fff;
        font-size: 4vw;
        margin-right: 4vw;

        display: flex;
        justify-content: center;
        align-items: center;
    }

    .wrapper .business li .business-info .business-info-star {
        display: flex;
        justify-content: space-between;
        align-items: center;
        margin-bottom: 2vw;
        font-size: 3.1vw;
    }

    .wrapper .business li .business-info .business-info-star .business-info-star-left {
        display: flex;
        align-items: center;
    }

    .wrapper .business li .business-info .business-info-star .business-info-star-left .fa-star {
        color: #FEC80E;

        margin-right: 0.5vw;
    }

    .wrapper .business li .business-info .business-info-star .business-info-star-left p {
        color: #666;
        margin-left: 1vw;
    }

    .wrapper .business li .business-info .business-info-star .business-info-star-right {
        background-color: #0097FF;
        color: #fff;
        font-size: 2.4vw;
        border-radius: 2px;
        padding: 0 0.6vw;
    }

    .wrapper .business li .business-info .business-info-delivery {
        display: flex;
        justify-content: space-between;
        align-items: center;
        margin-bottom: 2vw;

        color: #666;
        font-size: 3.1vw;
    }

    .wrapper .business li .business-info .business-info-explain {
        display: flex;
        align-items: center;
        margin-bottom: 3vw;
    }

    .wrapper .business li .business-info .business-info-explain div {
        border: solid 1px #DDD;
        font-size: 2.8vw;
        color: #666;
        border-radius: 3px;
        padding: 0 0.1vw;
    }

    .wrapper .business li .business-info .business-info-promotion {
        display: flex;
        justify-content: space-between;
        align-items: center;
        margin-bottom: 1.8vw;
    }

    .wrapper .business li .business-info .business-info-promotion .business-info-promotion-left
{
        display: flex;
        align-items: center;
    }


    .wrapper .business li .business-info .business-info-promotion .business-info-promotion-left
```

```
.business-info-promotion-left-incon {
        width: 4vw;
        height: 4vw;
        background-color: #70BC46;
        border-radius: 3px;
        font-size: 3vw;
        color: #fff;

        display: flex;
        justify-content: center;
        align-items: center;
    }

    .wrapper .business li .business-info .business-info-promotion .business-info-promotion-left p {
        color: #666;
        font-size: 3vw;
        margin-left: 2vw;
    }

    .wrapper .business li .business-info .business-info-promotion .business-info-promotion-right {
        display: flex;
        align-items: center;
        font-size: 2.5vw;
        color: #999;
    }

    .wrapper .business li .business-info .business-info-promotion .business-info-promotion-right p {
        margin-right: 2vw;
    }
</style>
```

4.3.2.BusinessList组件

```
<template>
    <div class="wrapper">

        <!-- header部分 -->
        <header>
            <p>商家列表</p>
        </header>

        <!-- 商家列表部分 -->
        <ul class="business">
            <li v-for="item in businessArr" @click="toBusinessInfo(item.businessId)">
                <div class="business-img">
                    <img :src="item.businessImg">

                    <div class="business-img-quantity" v-show="item.quantity>0">
```

```
{{item.quantity}}</div>
                </div>
                <div class="business-info">
                    <h3>{{item.businessName}}</h3>
                    <p>&#165;{{item.starPrice}}起送 | &#165;{{item.deliveryPrice}}配送</p>
                    <p>{{item.businessExplain}}</p>
                </div>
            </li>
        </ul>

        <!-- 底部菜单部分 -->
        <Footer></Footer>

    </div>
</template>

<script>
    import Footer from '../components/Footer.vue';

    export default{
        name:'BusinessList',
        data(){
            return {
                orderTypeId: this.$route.query.orderTypeId,
                businessArr:[],
                user:{}
            }
        },
        created() {
            this.user = this.$getSessionStorage('user');

            //根据orderTypeId查询商家信息
            this.$axios.post('BusinessController/listBusinessByOrderTypeId',this.$qs.stringify({
                orderTypeId:this.orderTypeId
            })).then(response=>{
                this.businessArr = response.data;
                //判断是否登录
                if(this.user!=null){
                    this.listCart();
                }
            }).catch(error=>{
                console.error(error);
            });
        },
        components:{
            Footer
        },
        methods:{
            listCart(){
                this.$axios.post('CartController/listCart',this.$qs.stringify({
                    userId:this.user.userId
                })).then(response=>{

                    let cartArr = response.data;
                    //遍历所有食品列表
                    for(let businessItem of this.businessArr){
                        businessItem.quantity = 0;
                        for(let cartItem of cartArr){
                            if(cartItem.businessId==businessItem.businessId){
                                businessItem.quantity += cartItem.quantity;
                            }
                        }
                    }
                    this.businessArr.sort();
                }).catch(error=>{
                    console.error(error);
                });
            },
            toBusinessInfo(businessId){
                this.$router.push({path:'/businessInfo',query:{businessId:businessId}});
            }
        }
    }
</script>

<style scoped>
    /****************** 总容器 ******************/
    .wrapper {
        width: 100%;
        height: 100%;
    }

    /****************** header部分 ******************/
    .wrapper header {
        width: 100%;
        height: 12vw;
        background-color: #0097FF;
        color: #fff;
        font-size: 4.8vw;

        position: fixed;
        left: 0;
        top: 0;
        z-index: 1000;

        display: flex;
        justify-content: center;
        align-items: center;
    }

    /****************** 商家列表部分 ******************/
    .wrapper .business {
        width: 100%;
        margin-top: 12vw;
        margin-bottom: 14vw;
    }
    .wrapper .business li {
        width: 100%;
        box-sizing: border-box;
        padding: 2.5vw;
        border-bottom: solid 1px #DDD;
        user-select: none;
        cursor: pointer;

        display: flex;
        align-items: center;
    }

    .wrapper .business li .business-img {
        /*这里设置为相当定位，成为business-img-quantity元素的父元素*/
        position: relative;
    }

    .wrapper .business li .business-img img {
        width: 20vw;
        height: 20vw;
    }

    .wrapper .business li .business-img .business-img-quantity {
        width: 5vw;
        height: 5vw;
        background-color: red;
        color: #fff;
        font-size: 3.6vw;
        border-radius: 2.5vw;

        display: flex;
        justify-content: center;
        align-items: center;

        /*设置成绝对定位，不占文档流空间*/
        position: absolute;
        right: -1.5vw;
        top: -1.5vw;
    }

    .wrapper .business li .business-info {
        margin-left: 3vw;
    }

    .wrapper .business li .business-info h3 {
        font-size: 3.8vw;
        color: #555;
    }

    .wrapper .business li .business-info p {
        font-size: 3vw;
        color: #888;

        margin-top: 2vw;
    }
</style>
```

4.3.3.BusinessInfo组件

```
<template>
    <div class="wrapper">

        <!-- header部分 -->
        <header>
            <p>商家信息</p>
        </header>

        <!-- 商家logo部分 -->
        <div class="business-logo">
            <img :src="business.businessImg">
        </div>

        <!-- 商家信息部分 -->
        <div class="business-info">
            <h1>{{business.businessName}}</h1>
            <p>&#165;{{business.starPrice}}起送 &#165;{{business.deliveryPrice}}配送</p>
            <p>{{business.businessExplain}}</p>
        </div>

        <!-- 食品列表部分 -->
        <ul class="food">
            <li v-for="(item,index) in foodArr">
                <div class="food-left">
                    <img :src="item.foodImg">
                    <div class="food-left-info">
                        <h3>{{item.foodName}}</h3>
                        <p>{{item.foodExplain}}</p>
                        <p>&#165;{{item.foodPrice}}</p>
                    </div>
                </div>
                <div class="food-right">
                    <div>
                        <i class="fa fa-minus-circle" @click="minus(index)" v-
show="item.quantity!=0"></i>
                    </div>
                    <p><span v-show="item.quantity!=0">{{item.quantity}}</span></p>
                    <div>
                        <i class="fa fa-plus-circle" @click="add(index)"></i>
                    </div>
                </div>
            </li>
        </ul>


        <!-- 购物车部分 -->
        <div class="cart">
            <div class="cart-left">
                <div class="cart-left-icon" :style="totalQuantity==0?'background-
color:#505051;':'background-color:#3190E8;'">
                    <i class="fa fa-shopping-cart"></i>
                    <div class="cart-left-icon-quantity" v-show="totalQuantity!=0">
{{totalQuantity}}</div>
                </div>
                <div class="cart-left-info">
                    <p>&#165;{{totalPrice}}</p>
                    <p>另需配送费{{business.deliveryPrice}}元</p>
                </div>
            </div>
            <div class="cart-right">
                <!-- 不够起送费 -->
                <div class="cart-right-item" v-show="totalSettle<business.starPrice"
style="background-color: #535356;cursor: default;">
                    &#165;{{business.starPrice}}起送
                </div>
                <!-- 达到起送费 -->
                <div class="cart-right-item" @click="toOrder" v-
show="totalSettle>=business.starPrice">
                    去结算
                </div>
            </div>
        </div>

    </div>
</template>

<script>
    export default {
        name: 'BusinessInfo',
        data() {
            return {
                businessId: this.$route.query.businessId,
                business: {},
                foodArr: [],
                user: {}
            }
        },
        created() {
            this.user = this.$getSessionStorage('user');

            //根据businessId查询商家信息
            this.$axios.post('BusinessController/getBusinessById', this.$qs.stringify({
                businessId: this.businessId
            })).then(response => {
                this.business = response.data;
            }).catch(error => {
                console.error(error);
            });
            //根据businessId查询所属食品信息
            this.$axios.post('FoodController/listFoodByBusinessId', this.$qs.stringify({
                businessId: this.businessId
            })).then(response => {
                this.foodArr = response.data;
                for (let i = 0; i < this.foodArr.length; i++) {
                    this.foodArr[i].quantity = 0;
                }

                //如果已登录，那么需要去查询购物车中是否已经选购了某个食品
                if (this.user != null) {
                    this.listCart();
                }
            }).catch(error => {
                console.error(error);
            });
        },
        methods: {
            listCart() {
                this.$axios.post('CartController/listCart', this.$qs.stringify({
                    businessId: this.businessId,
                    userId: this.user.userId
                })).then(response => {
                    let cartArr = response.data;
                    //遍历所有食品列表
                    for (let foodItem of this.foodArr) {
                        foodItem.quantity = 0;
                        for (let cartItem of cartArr) {
                            if (cartItem.foodId == foodItem.foodId) {
                                foodItem.quantity = cartItem.quantity;
                            }
                        }
                    }
                    this.foodArr.sort();
                }).catch(error => {
                    console.error(error);
                });
            },
            add(index) {
                //首先做登录验证
                if (this.user == null) {
                    this.$router.push({
                        path: '/login'
                    });
                    return;
                }

                if (this.foodArr[index].quantity == 0) {
                    //做insert
                    this.savaCart(index);
                } else {
                    //做update

                    this.updateCart(index, 1);
                }
            },
            minus(index) {
                //首先做登录验证
                if (this.user == null) {
                    this.$router.push({
                        path: '/login'
                    });
                    return;
                }

                if (this.foodArr[index].quantity > 1) {
                    //做update
                    this.updateCart(index, -1);
                } else {
                    //做delete
                    this.removeCart(index);
                }
            },
            savaCart(index) {
                this.$axios.post('CartController/saveCart', this.$qs.stringify({
                    businessId: this.businessId,
                    userId: this.user.userId,
                    foodId: this.foodArr[index].foodId
                })).then(response => {
                    if (response.data == 1) {
                        //此食品数量要更新为1;
                        this.foodArr[index].quantity = 1;
                        this.foodArr.sort();
                    } else {
                        alert('向购物车中添加食品失败！');
                    }
                }).catch(error => {
                    console.error(error);
                });
            },
            updateCart(index, num) {
                this.$axios.post('CartController/updateCart', this.$qs.stringify({
                    businessId: this.businessId,
                    userId: this.user.userId,
                    foodId: this.foodArr[index].foodId,
                    quantity: this.foodArr[index].quantity + num
                })).then(response => {
                    if (response.data == 1) {
                        //此食品数量要更新为1或-1;
                        this.foodArr[index].quantity += num;
                        this.foodArr.sort();
                    } else {
                        alert('向购物车中更新食品失败！');
                    }
                }).catch(error => {
                    console.error(error);

                });
            },
            removeCart(index) {
                this.$axios.post('CartController/removeCart', this.$qs.stringify({
                    businessId: this.businessId,
                    userId: this.user.userId,
                    foodId: this.foodArr[index].foodId
                })).then(response => {
                    if (response.data == 1) {
                        //此食品数量要更新为0；视图的减号和数量要消失
                        this.foodArr[index].quantity = 0;
                        this.foodArr.sort();
                    } else {
                        alert('从购物车中删除食品失败！');
                    }
                }).catch(error => {
                    console.error(error);
                });
            },
            toOrder() {
                this.$router.push({
                    path: '/orders',
                    query: {
                        businessId: this.business.businessId
                    }
                });
            }
        },
        computed: {
            //食品总价格
            totalPrice() {
                let total = 0;
                for (let item of this.foodArr) {
                    total += item.foodPrice * item.quantity;
                }
                return total;
            },
            //食品总数量
            totalQuantity() {
                let quantity = 0;
                for (let item of this.foodArr) {
                    quantity += item.quantity;
                }
                return quantity;
            },
            //结算总价格
            totalSettle() {
                return this.totalPrice + this.business.deliveryPrice;
            }
        }
    }
</script>


<style scoped>
```

```
/****************** 总容器 ******************/
.wrapper {
    width: 100%;
    height: 100%;
}

/****************** header部分 ******************/
.wrapper header {
    width: 100%;
    height: 12vw;
    background-color: #0097FF;
    color: #fff;
    font-size: 4.8vw;

    position: fixed;
    left: 0;
    top: 0;
    z-index: 1000;

    display: flex;
    justify-content: center;
    align-items: center;
}

/****************** 商家logo部分 ******************/
.wrapper .business-logo {
    width: 100%;
    height: 35vw;
    /*使用上外边距避开header部分*/
    margin-top: 12vw;

    display: flex;
    justify-content: center;
    align-items: center;
}

.wrapper .business-logo img {
    width: 40vw;
    height: 30vw;
    border-radius: 5px;
}

/****************** 商家信息部分 ******************/
.wrapper .business-info {
    width: 100%;
    height: 20vw;

    display: flex;
    flex-direction: column;
    justify-content: center;
    align-items: center;
}
```

```
.wrapper .business-info h1 {
    font-size: 5vw;
}

.wrapper .business-info p {
    font-size: 3vw;
    color: #666;
    margin-top: 1vw;
}

/****************** 食品列表部分 ******************/
.wrapper .food {
    width: 100%;
    /*使用下外边距避开footer部分*/
    margin-bottom: 14vw;
}

.wrapper .food li {
    width: 100%;
    box-sizing: border-box;
    padding: 2.5vw;
    user-select: none;

    display: flex;
    justify-content: space-between;
    align-items: center;
}

.wrapper .food li .food-left {
    display: flex;
    align-items: center;
}

.wrapper .food li .food-left img {
    width: 20vw;
    height: 20vw;
}

.wrapper .food li .food-left .food-left-info {
    margin-left: 3vw;
}

.wrapper .food li .food-left .food-left-info h3 {
    font-size: 3.8vw;
    color: #555;
}

.wrapper .food li .food-left .food-left-info p {
    font-size: 3vw;
    color: #888;
    margin-top: 2vw;
}
```

```
.wrapper .food li .food-right {
    width: 16vw;
    display: flex;
    justify-content: space-between;
    align-items: center;
}

.wrapper .food li .food-right .fa-minus-circle {
    font-size: 5.5vw;
    color: #999;
    cursor: pointer;
}

.wrapper .food li .food-right p {
    font-size: 3.6vw;
    color: #333;
}

.wrapper .food li .food-right .fa-plus-circle {
    font-size: 5.5vw;
    color: #0097EF;
    cursor: pointer;
}

/****************** 购物车部分 ******************/
.wrapper .cart {
    width: 100%;
    height: 14vw;

    position: fixed;
    left: 0;
    bottom: 0;

    display: flex;
}

.wrapper .cart .cart-left {
    flex: 2;
    background-color: #505051;
    display: flex;
}

.wrapper .cart .cart-left .cart-left-icon {
    width: 16vw;
    height: 16vw;
    box-sizing: border-box;
    border: solid 1.6vw #444;
    border-radius: 8vw;
    background-color: #3190E8;
    font-size: 7vw;
    color: #fff;


    display: flex;
```

```
        justify-content: center;
        align-items: center;

        margin-top: -4vw;
        margin-left: 3vw;

        position: relative;
    }

    .wrapper .cart .cart-left .cart-left-icon-quantity {
        width: 5vw;
        height: 5vw;
        border-radius: 2.5vw;
        background-color: red;
        color: #fff;
        font-size: 3.6vw;

        display: flex;
        justify-content: center;
        align-items: center;

        position: absolute;
        right: -1.5vw;
        top: -1.5vw;
    }

    .wrapper .cart .cart-left .cart-left-info p:first-child {
        font-size: 4.5vw;
        color: #fff;
        margin-top: 1vw;
    }

    .wrapper .cart .cart-left .cart-left-info p:last-child {
        font-size: 2.8vw;
        color: #AAA;
    }

    .wrapper .cart .cart-right {
        flex: 1;
    }

    /*达到起送费时的样式*/
    .wrapper .cart .cart-right .cart-right-item {
        width: 100%;
        height: 100%;
        background-color: #38CA73;
        color: #fff;
        font-size: 4.5vw;
        font-weight: 700;
        user-select: none;
        cursor: pointer;


        display: flex;
        justify-content: center;
        align-items: center;
    }

    /*不够起送费时的样式（只有背景色和鼠标样式的区别）*/
    /*
    .wrapper .cart .cart-right .cart-right-item{
        width: 100%;
        height: 100%;
        background-color: #535356;
        color: #fff;
        font-size: 4.5vw;
        font-weight: 700;
        user-select: none;

        display: flex;
        justify-content: center;
        align-items: center;
    }
    */
</style>
```

4.3.4.Order组件

```
<template>
    <div class="wrapper">

        <!-- header部分 -->
        <header>
            <p>确认订单</p>
        </header>

        <!-- 订单信息部分 -->
        <div class="order-info">
            <h5>订单配送至：</h5>
            <div class="order-info-address" @click="toUserAddress">
                <p>{{deliveryaddress!=null?deliveryaddress.address:'请选择送货地址'}}</p>
                <i class="fa fa-angle-right"></i>
            </div>
            <p>{{user.userName}}{{user.userSex | sexFilter}} {{user.userId}}</p>
        </div>

        <h3>{{business.businessName}}</h3>

        <!-- 订单明细部分 -->
        <ul class="order-detailed">
            <li v-for="item in cartArr">
                <div class="order-detailed-left">
                    <img :src="item.food.foodImg">

                    <p>{{item.food.foodName}} x {{item.quantity}}</p>
                </div>
                <p>&#165;{{item.food.foodPrice*item.quantity}}</p>
            </li>
        </ul>
        <div class="order-deliveryfee">
            <p>配送费</p>
            <p>&#165;{{business.deliveryPrice}}</p>
        </div>

        <!-- 合计部分 -->
        <div class="total">
            <div class="total-left">
                &#165;{{totalPrice}}
            </div>
            <div class="total-right" @click="toPayment">
                去支付
            </div>
        </div>
    </div>
</template>

<script>
    export default{
        name:'Orders',
        data(){
            return {
                businessId:this.$route.query.businessId,
                business:{},
                user:{},
                cartArr:[],
                deliveryaddress:{}
            }
        },
        created() {
            this.user = this.$getSessionStorage('user');
            this.deliveryaddress = this.$getLocalStorage(this.user.userId);

            //查询当前商家
            this.$axios.post('BusinessController/getBusinessById',this.$qs.stringify({
                businessId:this.businessId
            })).then(response=>{
                this.business = response.data;
            }).catch(error=>{
                console.error(error);
            });

            //查询当前用户在购物车中的当前商家食品列表
            this.$axios.post('CartController/listCart',this.$qs.stringify({
                userId:this.user.userId,
                businessId:this.businessId
            })).then(response=>{
                this.cartArr = response.data;

            }).catch(error=>{
                console.error(error);
            });
        },
        computed:{
            totalPrice(){
                let totalPrice = 0;
                for(let cartItem of this.cartArr){
                    totalPrice += cartItem.food.foodPrice*cartItem.quantity;
                }
                totalPrice += this.business.deliveryPrice;
                return totalPrice;
            }
        },
        filters:{
            sexFilter(value){
                return value==1?'先生':'女士';
            }
        },
        methods:{
            toUserAddress(){
                this.$router.push({path:'/userAddress',query:{businessId:this.businessId}});
            },
            toPayment(){
                if(this.deliveryaddress==null){
                    alert('请选择送货地址！');
                    return;
                }

                //创建订单
                this.$axios.post('OrdersController/createOrders',this.$qs.stringify({
                    userId:this.user.userId,
                    businessId:this.businessId,
                    daId:this.deliveryaddress.daId,
                    orderTotal:this.totalPrice
                })).then(response=>{
                    let orderId = response.data;
                    if(orderId>0){
                        this.$router.push({path:'/payment',query:{orderId:orderId}});
                    }else{
                        alert('创建订单失败！');
                    }
                }).catch(error=>{
                    console.error(error);
                });
            }
        }
    }
</script>

<style scoped>
    /****************** 总容器 ******************/
    .wrapper {

        width: 100%;
        height: 100%;
    }

    /****************** header部分 ******************/
    .wrapper header {
        width: 100%;
        height: 12vw;
        background-color: #0097FF;
        color: #fff;
        font-size: 4.8vw;

        position: fixed;
        left: 0;
        top: 0;
        z-index: 1000;

        display: flex;
        justify-content: center;
        align-items: center;
    }

    /****************** 订单信息部分 ******************/
    .wrapper .order-info {
        /*注意这里，不设置高，靠内容撑开。因为地址有可能折行*/
        width: 100%;
        margin-top: 12vw;
        background-color: #0097EF;
        box-sizing: border-box;
        padding: 2vw;
        color: #fff;
    }

    .wrapper .order-info h5 {
        font-size: 3vw;
        font-weight: 300;
    }

    .wrapper .order-info .order-info-address {
        width: 100%;
        display: flex;
        justify-content: space-between;
        align-items: center;

        font-weight: 700;
        user-select: none;
        cursor: pointer;
        margin: 1vw 0;
    }

    .wrapper .order-info .order-info-address p {
        width: 90%;
        font-size: 5vw;

    }
```

```
.wrapper .order-info .order-info-address i {
    font-size: 6vw;
}

.wrapper .order-info p {
    font-size: 3vw;
}

.wrapper h3 {
    box-sizing: border-box;
    padding: 3vw;
    font-size: 4vw;
    color: #666;
    border-bottom: solid 1px #DDD;
}

/****************** 订单明细部分 ******************/
.wrapper .order-detailed {
    width: 100%;
}

.wrapper .order-detailed li {
    width: 100%;
    height: 16vw;
    box-sizing: border-box;
    padding: 3vw;
    color: #666;

    display: flex;
    justify-content: space-between;
    align-items: center;
}

.wrapper .order-detailed li .order-detailed-left {
    display: flex;
    align-items: center;
}

.wrapper .order-detailed li .order-detailed-left img {
    width: 10vw;
    height: 10vw;
}

.wrapper .order-detailed li .order-detailed-left p {
    font-size: 3.5vw;
    margin-left: 3vw;
}

.wrapper .order-detailed li p {
    font-size: 3.5vw;
}
```

```
    .wrapper .order-deliveryfee {
        width: 100%;
        height: 16vw;
        box-sizing: border-box;
        padding: 3vw;
        color: #666;
        display: flex;
        justify-content: space-between;
        align-items: center;
        font-size: 3.5vw;
    }

    /****************** 订单合计部分 ******************/
    .wrapper .total {
        width: 100%;
        height: 14vw;

        position: fixed;
        left: 0;
        bottom: 0;

        display: flex;
    }

    .wrapper .total .total-left {
        flex: 2;
        background-color: #505051;
        color: #fff;
        font-size: 4.5vw;
        font-weight: 700;
        user-select: none;

        display: flex;
        justify-content: center;
        align-items: center;
    }

    .wrapper .total .total-right {
        flex: 1;
        background-color: #38CA73;
        color: #fff;
        font-size: 4.5vw;
        font-weight: 700;
        user-select: none;
        cursor: pointer;

        display: flex;
        justify-content: center;
        align-items: center;
    }
</style>
```

4.3.5.Payment组件

```
<template>
    <div class="wrapper">

        <!-- header部分 -->
        <header>
            <p>在线支付</p>
        </header>

        <!-- 订单信息部分 -->
        <h3>订单信息：</h3>
        <div class="order-info">
            <p>
                {{orders.business.businessName}}
                <i class="fa fa-caret-down" @click="detailetShow"></i>
            </p>
            <p>&#165;{{orders.orderTotal}}</p>
        </div>

        <!-- 订单明细部分 -->
        <ul class="order-detailet" v-show="isShowDetailet">
            <li v-for="item in orders.list">
                <p>{{item.food.foodName}} x {{item.quantity}}</p>
                <p>&#165;{{item.food.foodPrice*item.quantity}}</p>
            </li>
            <li>
                <p>配送费</p>
                <p>&#165;{{orders.business.deliveryPrice}}</p>
            </li>
        </ul>

        <!-- 支付方式部分 -->
        <ul class="payment-type">
            <li>
                <img src="../assets/alipay.png">
                <i class="fa fa-check-circle"></i>
            </li>
            <li>
                <img src="../assets/wechat.png">
            </li>
        </ul>
        <div class="payment-button">
            <button>确认支付</button>
        </div>

        <!-- 底部菜单部分 -->
        <Footer></Footer>
    </div>
</template>


<script>
```

```
    import Footer from '../components/Footer.vue';

    export default {
        name: 'Payment',
        data(){
            return {
                orderId:this.$route.query.orderId,
                orders:{
                    business:{}
                },
                isShowDetailet:false
            }
        },
        created() {
            this.$axios.post('OrdersController/getOrdersById',this.$qs.stringify({
                orderId:this.orderId
            })).then(response=>{
                this.orders = response.data;
            }).catch(error=>{
                console.error(error);
            });
        },
        mounted() {
            //这里的代码是实现：一旦路由到在线支付组件，就不能回到订单确认组件。
            //先将当前url添加到history对象中
            history.pushState(null,null,document.URL);
            //popstate事件能够监听history对象的变化
            window.onpopstate = () => {
                this.$router.push({path:'/index'});
            }
        },
        destroyed() {
            window.onpopstate = null;
        },
        methods:{
            detailetShow(){
                this.isShowDetailet = !this.isShowDetailet;
            }
        },
        components: {
            Footer
        }
    }
</script>

<style scoped>
    /****************** 总容器 ******************/
    .wrapper {
        width: 100%;
        height: 100%;
    }


    /****************** header部分 ******************/
    .wrapper header {
        width: 100%;
        height: 12vw;
        background-color: #0097FF;
        color: #fff;
        font-size: 4.8vw;

        position: fixed;
        left: 0;
        top: 0;
        z-index: 1000;

        display: flex;
        justify-content: center;
        align-items: center;
    }

    /****************** 订单信息部分 ******************/
    .wrapper h3 {
        margin-top: 12vw;
        box-sizing: border-box;
        padding: 4vw 4vw 0;

        font-size: 4vw;
        font-weight: 300;
        color: #999;
    }

    .wrapper .order-info {
        box-sizing: border-box;
        padding: 4vw;
        font-size: 4vw;
        color: #666;

        display: flex;
        justify-content: space-between;
        align-items: center;
    }

    .wrapper .order-info p:last-child {
        color: orangered;
    }

    /****************** 订单明细部分 ******************/
    .wrapper .order-detailet {
        width: 100%;
    }

    .wrapper .order-detailet li {
        width: 100%;
        box-sizing: border-box;
        padding: 1vw 4vw;
        display: flex;
        justify-content: space-between;
        align-items: center;
    }

    .wrapper .order-detailet li p {
        font-size: 3vw;
        color: #666;
    }

    /***************** 支付方式部分 *****************/
    .wrapper .payment-type {
        width: 100%;
    }

    .wrapper .payment-type li {
        width: 100%;
        box-sizing: border-box;
        padding: 4vw;

        display: flex;
        justify-content: space-between;
        align-items: center;
    }

    .wrapper .payment-type li img {
        width: 33vw;
        height: 8.9vw;
    }

    .wrapper .payment-type li .fa-check-circle {
        font-size: 5vw;
        color: #38CA73;
    }

    .wrapper .payment-button {
        width: 100%;
        box-sizing: border-box;
        padding: 4vw;
    }

    .wrapper .payment-button button {
        width: 100%;
        height: 10vw;
        border: none;
        /*去掉外轮廓线*/
        outline: none;
        border-radius: 4px;
        background-color: #38CA73;
        color: #fff;
    }
</style>
```

4.3.6.UserAddress组件

```
<template>
    <div class="wrapper">

        <!-- header部分 -->
        <header>
            <p>地址管理</p>
        </header>

        <!-- 地址列表部分 -->
        <ul class="addresslist">
            <li v-for="item in deliveryAddressArr">
                <div class="addresslist-left" @click="setDeliveryAddress(item)">
                    <h3>{{item.contactName}}{{item.contactSex | sexFilter}} {{item.contactTel}}</h3>
                    <p>{{item.address}}</p>
                </div>
                <div class="addresslist-right">
                    <i class="fa fa-edit" @click="editUserAddress(item.daId)"></i>
                    <i class="fa fa-remove" @click="removeUserAddress(item.daId)"></i>
                </div>
            </li>
        </ul>

        <!-- 新增地址部分 -->
        <div class="addbtn" @click="toAddUserAddress">
            <i class="fa fa-plus-circle"></i>
            <p>新增收货地址</p>
        </div>

        <!-- 底部菜单部分 -->
        <Footer></Footer>
    </div>
</template>

<script>
    import Footer from '../components/Footer.vue';

    export default{
        name:'UserAddress',
        data(){
            return {
                businessId:this.$route.query.businessId,
                user:{},
                deliveryAddressArr:[]
            }
        },
        created() {
            this.user = this.$getSessionStorage('user');
            this.listDeliveryAddressByUserId();
        },
        components:{
            Footer
        },
        filters:{
            sexFilter(value){
                return value==1?'先生':'女士';
            }
        },
        methods:{
            listDeliveryAddressByUserId(){
                //查询送货地址

this.$axios.post('DeliveryAddressController/listDeliveryAddressByUserId',this.$qs.stringify({
                    userId:this.user.userId
                })).then(response=>{
                    this.deliveryAddressArr = response.data;
                }).catch(error=>{
                    console.error(error);
                });
            },
            setDeliveryAddress(deliveryAddress){
                //把用户选择的默认送货地址存储到localStorage中
                this.$setLocalStorage(this.user.userId,deliveryAddress);
                this.$router.push({path:'/orders',query:{businessId:this.businessId}});
            },
            toAddUserAddress(){
                this.$router.push({path:'/addUserAddress',query:{businessId:this.businessId}});
            },
            editUserAddress(daId){
                this.$router.push({path:'/editUserAddress',query:
{businessId:this.businessId,daId:daId}});
            },
            removeUserAddress(daId){
                if(!confirm('确认要删除此送货地址吗？')){
                    return;
                }


this.$axios.post('DeliveryAddressController/removeDeliveryAddress',this.$qs.stringify({
                    daId:daId
                })).then(response=>{
                    if(response.data>0){
                        let deliveryAddress = this.$getLocalStorage(this.user.userId);
                        if(deliveryAddress!=null&&deliveryAddress.daId==daId){
                            this.$removeLocalStorage(this.user.userId);
                        }
                        this.listDeliveryAddressByUserId();
                    }else{
                        alert('删除地址失败！');
                    }

                }).catch(error=>{
                    console.error(error);
                });
            }
        }
    }
</script>

<style scoped>
    /*************** 总容器 ***************/
    .wrapper {
        width: 100%;
        height: 100%;
        background-color: #F5F5F5;
    }

    /*************** header ***************/
    .wrapper header {
        width: 100%;
        height: 12vw;
        background-color: #0097FF;
        display: flex;
        justify-content: space-around;
        align-items: center;
        color: #fff;
        font-size: 4.8vw;
        position: fixed;
        left: 0;
        top: 0;
        /*保证在最上层*/
        z-index: 1000;
    }

    /*************** addresslist ***************/
    .wrapper .addresslist {
        width: 100%;
        margin-top: 12vw;
        background-color: #fff;
    }

    .wrapper .addresslist li {
        width: 100%;
        box-sizing: border-box;
        border-bottom: solid 1px #DDD;
        padding: 3vw;

        display: flex;
    }

    .wrapper .addresslist li .addresslist-left {
        flex: 5;
        /*左边这块区域是可以点击的*/
        user-select: none;

        cursor: pointer;
    }

    .wrapper .addresslist li .addresslist-left h3 {
        font-size: 4.6vw;
        font-weight: 300;
        color: #666;
    }

    .wrapper .addresslist li .addresslist-left p {
        font-size: 4vw;
        color: #666;
    }

    .wrapper .addresslist li .addresslist-right {
        flex: 1;
        font-size: 5.6vw;
        color: #999;
        cursor: pointer;

        display: flex;
        justify-content: space-around;
        align-items: center;
    }

    /*************** 新增地址部分 ***************/
    .wrapper .addbtn {
        width: 100%;
        height: 14vw;
        border-top: solid 1px #DDD;
        border-bottom: solid 1px #DDD;
        background-color: #fff;
        margin-top: 4vw;

        display: flex;
        justify-content: center;
        align-items: center;

        font-size: 4.5vw;
        color: #0097FF;
        user-select: none;
        cursor: pointer;
    }

    .wrapper .addbtn p {
        margin-left: 2vw;
    }
</style>
```

4.3.7.AddUserAddress组件

```
<template>
    <div class="wrapper">

        <!-- header部分 -->
        <header>
            <p>新增送货地址</p>
        </header>

        <!-- 表单部分 -->
        <ul class="form-box">
            <li>
                <div class="title">
                    联系人：
                </div>
                <div class="content">
                    <input type="text" v-model="deliveryAddress.contactName" placeholder="联系人姓名">
                </div>
            </li>
            <li>
                <div class="title">
                    性别：
                </div>
                <div class="content" style="font-size: 3vw;">
                    <input type="radio" v-model="deliveryAddress.contactSex" value="1" style="width:6vw;height:3.2vw;">男
                    <input type="radio" v-model="deliveryAddress.contactSex" value="0" style="width:6vw;height:3.2vw;">女
                </div>
            </li>
            <li>
                <div class="title">
                    电话：
                </div>
                <div class="content">
                    <input type="tel" v-model="deliveryAddress.contactTel" placeholder="电话">
                </div>
            </li>
            <li>
                <div class="title">
                    收货地址：
                </div>
                <div class="content">
                    <input type="text" v-model="deliveryAddress.address" placeholder="收货地址">
                </div>
            </li>
        </ul>

        <div class="button-add">
            <button @click="addUserAddress">保存</button>
        </div>

        <!-- 底部菜单部分 -->
        <Footer></Footer>

    </div>
</template>

<script>
    import Footer from '../components/Footer.vue';

    export default {
        name: 'AddUserAddress',
        data() {
            return {
                businessId:this.$route.query.businessId,
                user:{},
                deliveryAddress:{
                    contactName:'',
                    contactSex:1,
                    contactTel:'',
                    address:''
                }
            }
        },
        created() {
            this.user = this.$getSessionStorage('user');
        },
        components: {
            Footer
        },
        methods: {
            addUserAddress(){
                if(this.deliveryAddress.contactName==''){
                    alert('联系人姓名不能为空！');
                    return;
                }
                if(this.deliveryAddress.contactTel==''){
                    alert('联系人电话不能为空！');
                    return;
                }
                if(this.deliveryAddress.address==''){
                    alert('联系人地址不能为空！');
                    return;
                }
                this.deliveryAddress.userId = this.user.userId;
                this.$axios.post('DeliveryAddressController/saveDeliveryAddress',
this.$qs.stringify(
                    this.deliveryAddress
                )).then(response => {
                    if(response.data>0){
                        this.$router.push({path:'/userAddress',query:
{businessId:this.businessId}});
                    }else{
                        alert('新增地址失败！');

                    }
                }).catch(error => {
                    console.error(error);
                });
            }
        }
    }
</script>

<style scoped>
    /*************** 总容器 ***************/
    .wrapper {
        width: 100%;
        height: 100%;
    }

    /*************** header ***************/
    .wrapper header {
        width: 100%;
        height: 12vw;
        background-color: #0097FF;
        display: flex;
        justify-content: space-around;
        align-items: center;
        color: #fff;
        font-size: 4.8vw;
        position: fixed;
        left: 0;
        top: 0;
        /*保证在最上层*/
        z-index: 1000;
    }

    /*************** （表单信息） ***************/
    .wrapper .form-box {
        width: 100%;
        margin-top: 12vw;
    }

    .wrapper .form-box li {
        box-sizing: border-box;
        padding: 4vw 3vw 0vw 3vw;
        display: flex;
    }

    .wrapper .form-box li .title {
        flex: 0 0 18vw;
        font-size: 3vw;
        font-weight: 700;
        color: #666;
    }

    .wrapper .form-box li .content {

        flex: 1;
        display: flex;
        align-items: center;
    }

    .wrapper .form-box li .content input {
        border: none;
        outline: none;
        width: 100%;
        height: 4vw;
        font-size: 3vw;
    }

    .wrapper .button-add {
        box-sizing: border-box;
        padding: 4vw 3vw 0vw 3vw;
    }

    .wrapper .button-add button {
        width: 100%;
        height: 10vw;
        font-size: 3.8vw;
        font-weight: 700;

        border: none;
        outline: none;
        border-radius: 4px;
        color: #fff;
        background-color: #38CA73;
    }

</style>
```

4.3.8.EditUserAddress组件

```
<template>
    <div class="wrapper">

        <!-- header部分 -->
        <header>
            <p>编辑送货地址</p>
        </header>

        <!-- 表单部分 -->
        <ul class="form-box">
            <li>
                <div class="title">
                    联系人:
                </div>

                <div class="content">
```

```
                    <input type="text" v-model="deliveryAddress.contactName" placeholder="联系人
姓名">
                </div>
            </li>
            <li>
                <div class="title">
                    性别:
                </div>
                <div class="content" style="font-size: 3vw;">
                    <input type="radio" v-model="deliveryAddress.contactSex" value="1"
style="width:6vw;height:3.2vw;" checked>男
                    <input type="radio" v-model="deliveryAddress.contactSex" value="0"
style="width:6vw;height:3.2vw;">女
                </div>
            </li>
            <li>
                <div class="title">
                    电话:
                </div>
                <div class="content">
                    <input type="tel" v-model="deliveryAddress.contactTel" placeholder="电话">
                </div>
            </li>
            <li>
                <div class="title">
                    收货地址:
                </div>
                <div class="content">
                    <input type="text" v-model="deliveryAddress.address" placeholder="收货地址">
                </div>
            </li>
        </ul>

        <div class="button-add">
            <button @click="editUserAddress">更新</button>
        </div>

        <!-- 底部菜单部分 -->
        <Footer></Footer>
    </div>
</template>

<script>
    import Footer from '../components/Footer.vue';

    export default {
        name: 'EditUserAddress',
        data() {
            return {
                businessId: this.$route.query.businessId,
                daId: this.$route.query.daId,
                user: {},

                deliveryAddress: {}
            }
        },
        created() {
            this.user = this.$getSessionStorage('user');

            this.$axios.post('DeliveryAddressController/getDeliveryAddressById',
this.$qs.stringify({
                daId:this.daId
            })).then(response => {
                this.deliveryAddress = response.data;
            }).catch(error => {
                console.error(error);
            });
        },
        components: {
            Footer
        },
        methods: {
            editUserAddress() {
                if (this.deliveryAddress.contactName == '') {
                    alert('联系人姓名不能为空！');
                    return;
                }
                if (this.deliveryAddress.contactTel == '') {
                    alert('联系人电话不能为空！');
                    return;
                }
                if (this.deliveryAddress.address == '') {
                    alert('联系人地址不能为空！');
                    return;
                }

                this.$axios.post('DeliveryAddressController/updateDeliveryAddress',
this.$qs.stringify(
                    this.deliveryAddress
                )).then(response => {
                    if (response.data > 0) {
                        this.$router.push({
                            path: '/userAddress',
                            query: {
                                businessId: this.businessId
                            }
                        });
                    } else {
                        alert('更新地址失败！');
                    }
                }).catch(error => {
                    console.error(error);
                });
            }
        }
    }

</script>
```

```
<style scoped>
    /*************** 总容器 ***************/
    .wrapper {
        width: 100%;
        height: 100%;
    }

    /*************** header ***************/
    .wrapper header {
        width: 100%;
        height: 12vw;
        background-color: #0097FF;
        display: flex;
        justify-content: space-around;
        align-items: center;
        color: #fff;
        font-size: 4.8vw;
        position: fixed;
        left: 0;
        top: 0;
        /*保证在最上层*/
        z-index: 1000;
    }

    /*************** （表单信息） ***************/
    .wrapper .form-box {
        width: 100%;
        margin-top: 12vw;
    }

    .wrapper .form-box li {
        box-sizing: border-box;
        padding: 4vw 3vw 0vw 3vw;
        display: flex;
    }

    .wrapper .form-box li .title {
        flex: 0 0 18vw;
        font-size: 3vw;
        font-weight: 700;
        color: #666;
    }

    .wrapper .form-box li .content {
        flex: 1;

        display: flex;
        align-items: center;
    }

    .wrapper .form-box li .content input {

        border: none;
        outline: none;
        width: 100%;
        height: 4vw;
        font-size: 3vw;
    }

    .wrapper .button-add {
        box-sizing: border-box;
        padding: 4vw 3vw 0vw 3vw;
    }

    .wrapper .button-add button {
        width: 100%;
        height: 10vw;
        font-size: 3.8vw;
        font-weight: 700;

        border: none;
        outline: none;
        border-radius: 4px;
        color: #fff;
        background-color: #38CA73;
    }
</style>
```

4.3.9.OrderList组件

```
<template>
    <div class="wrapper">

        <!-- header部分 -->
        <header>
            <p>我的订单</p>
        </header>

        <!-- 订单列表部分 -->
        <h3>未支付订单信息：</h3>
        <ul class="order">
            <li v-for="item in orderArr" v-if="item.orderState==0">
                <div class="order-info">
                    <p>
                        {{item.business.businessName}}
                        <i class="fa fa-caret-down" @click="detailetShow(item)"></i>
                    </p>
                    <div class="order-info-right">
                        <p>&#165;{{item.orderTotal}}</p>
                        <div class="order-info-right-icon">去支付</div>
                    </div>
                </div>

                <ul class="order-detailet" v-show="item.isShowDetailet">
```

```
                    <li v-for="odItem in item.list">
                        <p>{{odItem.food.foodName}} x {{odItem.quantity}}</p>
                        <p>&#165;{{odItem.food.foodPrice*odItem.quantity}}</p>
                    </li>
                    <li>
                        <p>配送费</p>
                        <p>&#165;{{item.business.deliveryPrice}}</p>
                    </li>
                </ul>
            </li>
        </ul>

        <h3>已支付订单信息：</h3>
        <ul class="order">
            <li v-for="item in orderArr" v-if="item.orderState==1">
                <div class="order-info">
                    <p>
                        {{item.business.businessName}}
                        <i class="fa fa-caret-down" @click="detailetShow(item)"></i>
                    </p>
                    <div class="order-info-right">
                        <p>&#165;{{item.orderTotal}}</p>
                    </div>
                </div>
                <ul class="order-detailet" v-show="item.isShowDetailet">
                    <li v-for="odItem in item.list">
                        <p>{{odItem.food.foodName}} x {{odItem.quantity}}</p>
                        <p>&#165;{{odItem.food.foodPrice*odItem.quantity}}</p>
                    </li>
                    <li>
                        <p>配送费</p>
                        <p>&#165;{{item.business.deliveryPrice}}</p>
                    </li>
                </ul>
            </li>
        </ul>

        <!-- 底部菜单部分 -->
        <Footer></Footer>

    </div>
</template>

<script>
    import Footer from '../components/Footer.vue';

    export default{
        name:'OrderList',
        data(){
            return {
                orderArr:[],
                user:{}

            }
        },
        created() {
            this.user = this.$getSessionStorage('user');

            this.$axios.post('OrdersController/listOrdersByUserId',this.$qs.stringify({
                userId:this.user.userId
            })).then(response=>{
                let result = response.data;
                for(let orders of result){
                    orders.isShowDetailet = false;
                }
                this.orderArr = result;
            }).catch(error=>{
                console.error(error);
            });
        },
        methods:{
            detailetShow(orders){
                orders.isShowDetailet = !orders.isShowDetailet;
            }
        },
        components:{
            Footer
        }
    }
</script>

<style scoped>
    /****************** 总容器 ******************/
    .wrapper {
        width: 100%;
        height: 100%;
    }

    /****************** header部分 ******************/
    .wrapper header {
        width: 100%;
        height: 12vw;
        background-color: #0097FF;
        color: #fff;
        font-size: 4.8vw;

        position: fixed;
        left: 0;
        top: 0;
        z-index: 1000;

        display: flex;
        justify-content: center;
        align-items: center;
    }


    /****************** 历史订单列表部分 ******************/
    .wrapper h3 {
        margin-top: 12vw;
        box-sizing: border-box;
        padding: 4vw;
        font-size: 4vw;
        font-weight: 300;
        color: #999;
    }

    .wrapper .order {
        width: 100%;
    }

    .wrapper .order li {
        width: 100%;
    }

    .wrapper .order li .order-info {
        box-sizing: border-box;
        padding: 2vw 4vw;
        font-size: 4vw;
        color: #666;

        display: flex;
        justify-content: space-between;
        align-items: center;
    }

    .wrapper .order li .order-info .order-info-right {
        display: flex;
    }

    .wrapper .order li .order-info .order-info-right .order-info-right-icon {
        background-color: #f90;
        color: #fff;
        border-radius: 3px;
        margin-left: 2vw;
        user-select: none;
        cursor: pointer;
    }

    .wrapper .order li .order-detailet {
        width: 100%;
    }

    .wrapper .order li .order-detailet li {
        width: 100%;
        box-sizing: border-box;
        padding: 1vw 4vw;
        color: #666;
        font-size: 3vw;


        display: flex;
        justify-content: space-between;
        align-items: center;
    }
</style>
```

4.3.10.Login组件

```
<template>
    <div class="wrapper">

        <!-- header部分 -->
        <header>
            <p>用户登陆</p>
        </header>

        <!-- 表单部分 -->
        <ul class="form-box">
            <li>
                <div class="title">
                    手机号码：
                </div>
                <div class="content">
                    <input type="text" v-model="userId" placeholder="手机号码">
                </div>
            </li>
            <li>
                <div class="title">
                    密码：
                </div>
                <div class="content">
                    <input type="password" v-model="password" placeholder="密码">
                </div>
            </li>
        </ul>

        <div class="button-login">
            <button @click="login">登陆</button>
        </div>
        <div class="button-register">
            <button @click="register">去注册</button>
        </div>

        <!-- 底部菜单部分 -->
        <Footer></Footer>
    </div>
</template>

<script>
    import Footer from '../components/Footer.vue';
    export default{
        name:'Login',
        data(){
            return {
                userId:'',
                password:''
            }
        },
        methods:{
            login(){
                if(this.userId==''){
                    alert('手机号码不能为空！');
                    return;
                }
                if(this.password==''){
                    alert('密码不能为空！');
                    return;
                }

                //登录请求
                this.$axios.post('UserController/getUserByIdByPass',this.$qs.stringify({
                    userId:this.userId,
                    password:this.password
                })).then(response=>{
                    let user = response.data;
                    if(user==null||user==''){
                        alert('用户名或密码不正确！');
                    }else{
                        //sessionstorage有容量限制，为了防止数据溢出，所以不将userImg数据放入session中

                        user.userImg = '';
                        this.$setSessionStorage('user',user);
                        this.$router.go(-1);
                    }
                }).catch(error=>{
                    console.error(error);
                });
            },
            register(){
                this.$router.push({path:'register'});
            }
        },
        components:{
            Footer
        }
    }
</script>

<style scoped>
    /****************** 总容器 ******************/
    .wrapper {
        width: 100%;

        height: 100%;
```

```
}

/****************** header部分 ******************/
.wrapper header {
    width: 100%;
    height: 12vw;
    background-color: #0097FF;
    color: #fff;
    font-size: 4.8vw;

    position: fixed;
    left: 0;
    top: 0;
    z-index: 1000;

    display: flex;
    justify-content: center;
    align-items: center;
}

/****************** 表单部分 ******************/
.wrapper .form-box {
    width: 100%;
    margin-top: 12vw;
}

.wrapper .form-box li {
    box-sizing: border-box;
    padding: 4vw 3vw 0 3vw;
    display: flex;
    align-items: center;
}

.wrapper .form-box li .title {
    flex: 0 0 18vw;
    font-size: 3vw;
    font-weight: 700;
    color: #666;
}

.wrapper .form-box li .content {
    flex: 1;
}

.wrapper .form-box li .content input {
    border: none;
    outline: none;
    width: 100%;
    height: 4vw;
    font-size: 3vw;
}


.wrapper .button-login {
    width: 100%;
    box-sizing: border-box;
    padding: 4vw 3vw 0 3vw;
}

.wrapper .button-login button {
    width: 100%;
    height: 10vw;
    font-size: 3.8vw;
    font-weight: 700;
    color: #fff;
    background-color: #38CA73;
    border-radius: 4px;

    border: none;
    outline: none;
}

.wrapper .button-register {
    width: 100%;
    box-sizing: border-box;
    padding: 4vw 3vw 0 3vw;
}

.wrapper .button-register button {
    width: 100%;
    height: 10vw;
    font-size: 3.8vw;
    font-weight: 700;
    /*与上面登陆按钮不同的只有颜色、背景色、边框不同*/
    color: #666;
    background-color: #EEE;
    border: solid 1px #DDD;
    border-radius: 4px;

    border: none;
    outline: none;
}

/****************** 底部菜单部分 ******************/
.wrapper .footer {
    width: 100%;
    height: 14vw;
    border-top: solid 1px #DDD;
    background-color: #fff;

    position: fixed;
    left: 0;
    bottom: 0;

    display: flex;
    justify-content: space-around;

    align-items: center;
}

.wrapper .footer li {
    display: flex;
    flex-direction: column;
    justify-content: center;
    align-items: center;

    color: #999;
    user-select: none;
    cursor: pointer;
}

.wrapper .footer li p {
    font-size: 2.8vw;
}

.wrapper .footer li i {
    font-size: 5vw;
}
</style>
```

4.3.11.Register组件

```
<template>
    <div class="wrapper">

        <!-- header部分 -->
        <header>
            <p>用户注册</p>
        </header>

        <!-- 表单部分 -->
        <ul class="form-box">
            <li>
                <div class="title">
                    手机号码：
                </div>
                <div class="content">
                    <input type="text" @blur="checkUserId" v-model="user.userId" placeholder="手机号码">
                </div>
            </li>
            <li>
                <div class="title">
                    密码：
                </div>
                <div class="content">
                    <input type="password" v-model="user.password" placeholder="密码">

                </div>
            </li>
            <li>
                <div class="title">
                    确认密码:
                </div>
                <div class="content">
                    <input type="password" v-model="confirmPassword" placeholder="确认密码">
                </div>
            </li>
            <li>
                <div class="title">
                    用户名称:
                </div>
                <div class="content">
                    <input type="text" v-model="user.userName" placeholder="用户名称">
                </div>
            </li>
            <li>
                <div class="title">
                    性别:
                </div>
                <div class="content" style="font-size: 3vw;">
                    <input type="radio" v-model="user.userSex" value="1"
style="width:6vw;height: 3.2vw;">男
                    <input type="radio" v-model="user.userSex" value="0"
style="width:6vw;height: 3.2vw;">女
                </div>
            </li>
        </ul>

        <div class="button-login">
            <button @click="register">注册</button>
        </div>

        <!-- 底部菜单部分 -->
        <Footer></Footer>
    </div>
</template>

<script>
    import Footer from '../components/Footer.vue';

    export default {
        name: 'Register',
        data() {
            return {
                user:{
                    userId:'',
                    password:'',
                    userName:'',
                    userSex:1
                },

                confirmPassword:''
            }
        },
        methods: {
            checkUserId(){
                this.$axios.post('UserController/getUserById', this.$qs.stringify({
                    userId: this.user.userId,
                })).then(response => {
                    if(response.data==1){
                        this.user.userId = '';
                        alert('此手机号码已存在！')
                    }
                }).catch(error => {
                    console.error(error);
                });
            },
            register() {
                if (this.user.userId == '') {
                    alert('手机号码不能为空！');
                    return;
                }
                if (this.user.password == '') {
                    alert('密码不能为空！');
                    return;
                }
                if (this.user.password != this.confirmPassword) {
                    alert('两次输入的密码不一致！');
                    return;
                }
                if (this.user.userName == '') {
                    alert('用户名不能为空！');
                    return;
                }

                //注册请求
                this.$axios.post('UserController/saveUser', this.$qs.stringify(
                    this.user
                )).then(response => {
                    if(response.data>0){
                        alert('注册成功！');
                        this.$router.go(-1);
                    }else{
                        alert('注册失败！');
                    }
                }).catch(error => {
                    console.error(error);
                });
            }
        },
        components: {
            Footer
        }
    }

</script>
```

```
<style scoped>
    /****************** 总容器 ******************/
    .wrapper {
        width: 100%;
        height: 100%;
    }

    /****************** header部分 ******************/
    .wrapper header {
        width: 100%;
        height: 12vw;
        background-color: #0097FF;
        color: #fff;
        font-size: 4.8vw;

        position: fixed;
        left: 0;
        top: 0;
        z-index: 1000;

        display: flex;
        justify-content: center;
        align-items: center;
    }

    /****************** 表单部分 ******************/
    .wrapper .form-box {
        width: 100%;
        margin-top: 12vw;
    }

    .wrapper .form-box li {
        box-sizing: border-box;
        padding: 4vw 3vw 0 3vw;
        display: flex;
        align-items: center;
    }

    .wrapper .form-box li .title {
        flex: 0 0 18vw;
        font-size: 3vw;
        font-weight: 700;
        color: #666;
    }

    .wrapper .form-box li .content {
        flex: 1;
    }

    .wrapper .form-box li .content input {
        border: none;

        outline: none;
        width: 100%;
        height: 4vw;
        font-size: 3vw;
    }

    .wrapper .button-login {
        width: 100%;
        box-sizing: border-box;
        padding: 4vw 3vw 0 3vw;
    }

    .wrapper .button-login button {
        width: 100%;
        height: 10vw;
        font-size: 3.8vw;
        font-weight: 700;
        color: #fff;
        background-color: #38CA73;
        border-radius: 4px;

        border: none;
        outline: none;
    }

    .wrapper .button-register {
        width: 100%;
        box-sizing: border-box;
        padding: 4vw 3vw 0 3vw;
    }

    .wrapper .button-register button {
        width: 100%;
        height: 10vw;
        font-size: 3.8vw;
        font-weight: 700;
        color: #666;
        background-color: #EEE;
        border-radius: 4px;

        border: none;
        outline: none;
        border: solid 1px #DDD;
    }
</style>
```

4.3.12.Footer共通组件

```
<template>
    <ul class="footer">

        <li @click="toIndex">
```

```
            <i class="fa fa-home"></i>
            <p>首页</p>
        </li>
        <li>
            <i class="fa fa-compass"></i>
            <p>发现</p>
        </li>
        <li @click="toOrderList">
            <i class="fa fa-file-text-o"></i>
            <p>订单</p>
        </li>
        <li>
            <i class="fa fa-user-o"></i>
            <p>我的</p>
        </li>
    </ul>
</template>

<script>
    export default{
        name:'Footer',
        methods:{
            toIndex(){
                this.$router.push({path:'/index'});
            },
            toOrderList(){
                this.$router.push({path:'/orderList'});
            }
        }
    }
</script>

<style>
    .wrapper .footer {
        width: 100%;
        height: 14vw;
        border-top: solid 1px #DDD;
        background-color: #fff;

        position: fixed;
        left: 0;
        bottom: 0;

        display: flex;
        justify-content: space-around;
        align-items: center;
    }

    .wrapper .footer li {
        /*li本身的尺寸完全由内容撑起*/
        display: flex;
        flex-direction: column;

        justify-content: center;
        align-items: center;

        color: #999;
        user-select: none;
        cursor: pointer;
    }

    .wrapper .footer li p {
        font-size: 2.8vw;
    }

    .wrapper .footer li i {
        font-size: 5vw;
    }
</style>
```