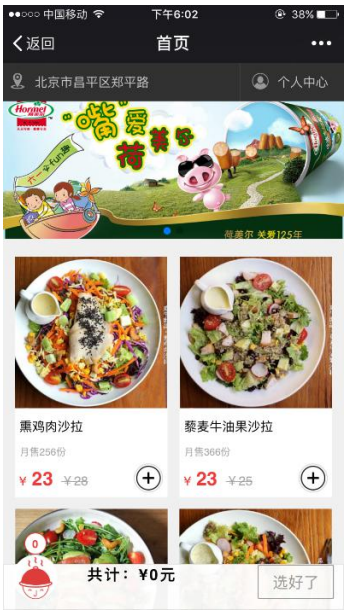
高仿”焦耳外卖”Web App

1. 准备

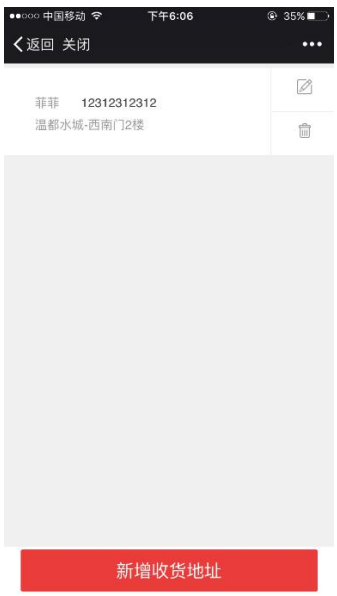
1.1. 项目描述

- 此项目为高仿微信公众号外卖 APP: 焦耳外卖
- 功能模块包括: 首页,登陆,购物车,地址管理,地图定位,下单等功能
- 编码实现分为前端编码与后端编码(全栈)
- 前端以 angularjs 技术栈为核心开发的 SPA
- 后端采用 node(Express) + MongDB(mongoose)技术组合
- 使用模块化编码与自动化构建

1.2. 项目功能界面

<table>
<tr><td>01_首页</td><td>02_登陆</td><td>03_个人中心</td></tr>
<tr><td> 
</td><td>
</td><td>
</td></tr>
<tr><td>04_意见反馈</td><td>05_地址管理</td><td>06_地址添加</td></tr>
</table>

07_地图选择位置



08_地址更新

09_订单确认

10_订单详情

11_用户订单列表

●●○○○ 中国移动
下午6:08
34%
返回 关闭
订单详情
已完成
预计送达：19:07
已付款
商家已接单
确认送达
订单详情
藜麦牛油果沙拉 X2 23元
熏鸡肉沙拉 X1 23元
配送费用 5元
费用合计 74元
配送地址：温都水城-西南门2楼
收货电话：12312312312
退单
确认送达

●●○○○ 中国移动
下午6:06
35%
返回
订单管理
订单号：5821cc3398877754d4199c70
状态：已完成
黑椒牛肉沙拉 X2 38元
吞拿鱼沙拉 X2 21元
配送地址：温都水城-西南门2楼
配送费用：5元
送餐时间：11-08 22:14
优惠减免：0 元
下单时间：
实付：118元
订单号：5825791c035fd62cc829c192

1.3. 技术选型

前台数据展现
angular
angular-router
angular-messages
后台服务
node
express
mongoDB
mongoose
前后台交互
接口文档
angular自带ajax
模块化
AMD
requirejs
项目构建
grunt
grunt相关插件
滑动库
swiper
编码规范检查
jslint

1.4. 前端路由

/home
首页
templateUrl：'js/templates/home.html'
controller：'HomeCtrl'
/personal
个人中心
templateUrl：'js/templates/personalCenter.html'
controller：'PersonalCtrl'
/login
登陆
templateUrl：'js/templates/login.html'
controller：'LoginCtrl'
/feedback
意见反馈
templateUrl：'js/templates/feedback.html'
controller：'FeedbackCtrl'
/addr_manage
地址管理
templateUrl：'js/templates/addrManage.html'
controller：'AddrManageCtrl'
/add_new_addr
新增/更新地址
templateUrl：'js/templates/addNewAddr.html'
controller：'AddNewAddrCtrl'
/choose_oordinate
选择地图地址
templateUrl：'js/templates/chooseCoordinate.html'
controller：'ChooseCoordinateCtrl'
/location_addr
定位当前地址
templateUrl：'js/templates/locationAddr.html'
controller：'LocationAddrCtrl'
/order_confirm
订单确认
templateUrl：'js/templates/orderConfirm.html'
controller：'OrderConfirmCtrl'

1.5. API 接口

2. 应用开发详解

2.1. 搭环境

1) 安装 node 与 mongodb，启动 mongodb 服务

- 下载并安装 node

```
* node -v
```

- 下载并安装 mongodb

```
* mongo --version
```

- 启动 mongodb 服务并测试

```
* win10 可能需要单独使用 bat 文件启动
```

- mongodb 常用命令

```
* mongo : 连接数据库
* show dbs :  显示所有的数据库
* use atguigu_o2o : 使用某个库
* show collections : 显示库中的所有集合(表)
* db.users.find() : 查询集合中所有的文档(记录)
* db.dropDatabase() : 删除指定的库
```

- mongodb2 个重要概念

* collection(集合)：多条记录的集合　相当于表，查询得到对象的数组
* document(文档)：一条记录，查询得到对象

2) WebStorm 打开最终版项目源码工程

- server 部分

```
|- bin
   |- www : 项目启动文件
```

```
|- db : 数据模块文件夹
|- node_modules : 依赖模块
|- public : client 资源文件夹
|- routes : 路由模块文件夹
|- util : 工具模块文件夹
|- views : 服务器端模板页面文件夹
|- app.js : 项目的主 js
|- package.json : 项目应用包描述文件
```

- client 部分

```
public
|- css : css 文件夹
|- img : 图片文件夹
|- js
    |- libs : 库 js
    |- app : angular module 模块 js
    |- controllers : 控制器模块 js
    |- routes : 路由模块 js
    |- services : 服务模块 js
    |- util : 工具模块 js
    |- templates : 模板 html
|- index.html : 主页面
```

- 文档与笔记

```
|- 笔记  : 笔记文件夹
|- 接口文档.md : 前后台交互接口文档
```

3) 导入项目的初始化数据库数据

- 命令行窗口进入: db/mongoDB_data_file
- 运行'导入数据库的命令.txt'中的所有命令: 初始化数据库数据

4) 启动运行项目

- 命令行运行
  - node bin/www
  - npm run start
  - npm start
- 通过工具运行 :
  - 右键 www 文件选择 run/debug

5) 访问体验项目功能

- pc 访问:

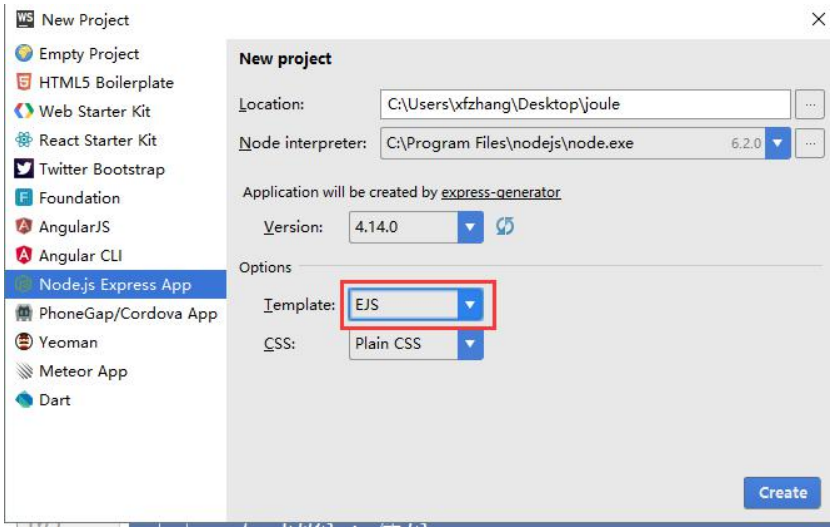
```
* 自己访问: localhost:3000
* 他人访问: ip:3000
```

- 手机访问:

```
* 电脑连接外网
* 电脑共享出一个热点
* 手机联上电脑热点
* 查看电脑的虚拟 IP
* 手机浏览器访问: http://ip:4000
```

2.2. 创建新项目及其整体结构

- 创建一个 node 项目



- 使用 SVN 对项目进行版本控制
  * 忽略.idea 和 node_modules
- 分析项目 client 端的整体结构设计
- 创建 client 端的整体文件结构

```
public
  |- css: css 文件夹, 基础 css 和各个模板页面都对应一个 css 文件
  |- img: 图片文件夹
  |- js
    |- libs: 库 js
    |- app: angular module 模块 js
    |- controllers: 控制器模块 js
    |- routes: 路由模块 js
    |- services: 服务模块 js
```

```
        |- utils: 工具模块 js
        |- templates: 模板 html
    |- index.html: 主页面
```

* 导入所有依赖的 js 库

```
 * angular.js : angular 核心库
 * angular-messages.js : angular 表单验证插件
 * angular-route.js : angular 路由插件
 * require.js : AMD 模块化库
```

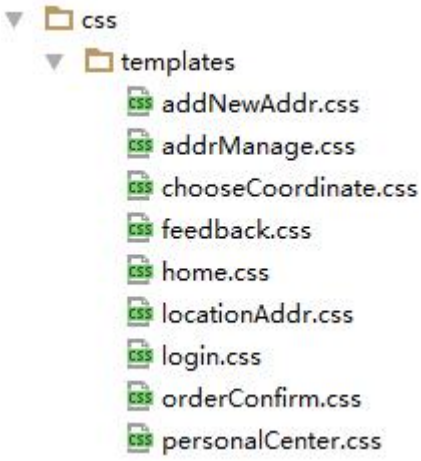
2.3. 使用 require.js 实现模块化编码

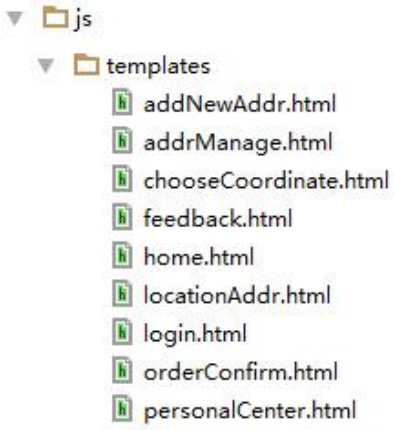
1) 拷贝静态资源到对应目录

```
css--------------*.css
img--------------*.jpng/png
js/templates-----*.html
```

2) 拆分页面

- 将 html 中的<style>提取出来作为一个单独的 css 文件，放入 css/templates/xxx.css
- 去除 html 中整体标签，只留下<body 标签的内容
- 在 html 的顶部引入对应的 css 样式





- css/templates/home.css

```
@media only screen and  (min-width: 320px) and  (max-width: 480px) {
    html{
        font-size: 14px;
    }
}
body{
```

```
    padding: 3rem 0 3.5rem 0;
}

/* 头部 **/
header{
    position: fixed;

    top: 0;
    left: 0;
    display: table;
    width: 100%;
    padding : 0 5px;
    background: #282828;
    background: hsla(0,0%,16%,.95);
    color: #ababab;
    font-size: 1rem;
    z-index: 999;
}
header::before {
    content: '';
    position: absolute;
    top: 0; right: 0; bottom: 0; left: 0;
    -webkit-filter: blur(3rem);
    filter: blur(3rem);
    z-index: -1;
}
header .addr,
header .user{
    display: table-cell;
    vertical-align: middle;
    /*padding: 0 20px;*/
    height: 3rem;
    overflow: hidden;
}
header .addr{
    width: 70%;
    border-right: 1px solid #3e3e3e;
}
header .addr a{
    display: inline-block;
    overflow: hidden;
    text-overflow: ellipsis;
    white-space: nowrap;
    width: 70%;
```

```
}
header .user{
    width: 30%;
    text-align: center;
}
header a,
header a:link,
header a:visited{
    color: #ccc;
}
.icon{
    display: inline-block;
    width: 20px;
    height: 20px;
    vertical-align: middle;
    margin: -5px 5px 0 0;
}
.icon-addr{
    background: url(../../img/addr.png);
    background-size: 100% auto;
}
.icon-user{
    background: url(../../img/touxiang.png);
    background-size: 100% auto;
}
/* 轮播 */
.banner{
    position: relative;
    line-height: 0;
    width: 100%;
}
.banner img{
    width: 100%;
}
.swiper-pagination {
    position: absolute;
    bottom: 5px;
    text-align: center;
    width: 100%;
    z-index: 5;
}
.swiper-pagination-bullet-active {
    opacity: 1;
    background: #eb3c3c;
```

```
}
.swiper-pagination-bullet {
    margin: 0 3px;
}


/* 菜品列表 **/
.meals{
    padding: 0.94rem 0.47rem;
    margin-bottom: 1rem;
}
.meals ul{

}
.meals li{
    width: 50%;
    padding:  0.47rem ;
    float: left;
}

/* 菜品项 **/
.meal-item{
    background: #fff;
}

.meals .img{
    display: block;
    width: 100%;
    line-height: 0;
    border: 0;
}
.meals .img img{
    width: 100%;
}
/* 菜品信息 **/
.meal-name,
.sales,
.buy{
    margin: 10px 5px 0 5px  ;
}
.meal-name{
    overflow: hidden;
    text-overflow: ellipsis;
    white-space: nowrap;
```

```
    font-size: 1rem;
}
.meals .sales{
    color: #a3a3a3;
    font-size: 0.8rem;
}
.buy{
    display: table;
    width: 100%;
    vertical-align: middle;
    margin: 0;
    padding: 5px;
    color: #eb3c3c;
    font-size: 1rem;
}
.meals .price,
.option-cart{
    display: table-cell;
    vertical-align: middle;
}
.meals .price s{
    margin: 0 0 0 5px;
    color: #a3a3a3;
}
.meals .close{
    color: #a3a3a3;
}

/* 购物车加减按钮 */
.btn-cart{
    display: table;
    padding: 1px;
    height: 2.06rem;
    -moz-border-radius:2.06rem;
    -webkit-border-radius: 2.06rem;
    border-radius: 2.06rem;
    border: 1px solid #dcdcdc;
    font-weight: bold;
    color: #000;
}
.btn-cart .jia,
.btn-cart .jian,
.btn-cart .count{
    display: table-cell;
```

```
    vertical-align: middle;
}

.btn-cart .count{
    font-size: 1rem;
    padding: 0 3px;
}

.btn-cart a,
.jia-fly{
    width: 2rem;
    height:2rem;
    -moz-border-radius: 2rem;
    -webkit-border-radius:2rem;
    border-radius: 2rem;
    background: #fff;
    /*border: 1px solid #000;*/
    text-align: center;
    font-size: 1.8rem;
    line-height: 1.6rem;
    font-weight: normal;
    background:
url(
AsTAAALEwEAmpwYAAA7U2lUWHRYTUw6Y29tLmFkb2JlLnhtcAAAAAAAPD94cGFja2V0IGJlZ2luPSLvu78i
IGlkPSJXNU0wTXBDZWhpSHpyZVN6TlRjemtjOWQiPz4KPHg6eG1wbWV0YSB4bWxuczp4PSJhZG9iZTpuczp
tZXRhLyIgeDp4bXB0az0iQWRvYmUgWE1QIENvcmUgNS41LWMwMTQgNzkuMTUxNDgxLCAyMDEzLzAzLzEzLT
EyOjA5OjE1ICAgICAgICAiPgogICA8cmRmOlJERiB4bWxuczpyZGY9Imh0dHA6Ly93d3cudzMub3JnLzE5O
TkvMDIvMjItcmRmLXN5bnRheC1ucyMiPgogICAgICA8cmRmOkRlc2NyaXB0aW9uIHJkZjphYm91dD0iIgog
ICAgICAgICAgICB4bWxuczp4bXBNTT0iaHR0cDovL25zLmFkb2JlLmNvbS94YXAvMS4wL21tLyIKICAgICA
gICAgICAgICAgeG1sbnM6c3RSZWY9Imh0dHA6Ly9ucy5hZG9iZS5jb20veGFwLzEuMC9zVHlwZS9SZXNvdXJjZV
JlZiMiCiAgICAgICAgICAgIHhtbG5zOnN0RXZ0PSJodHRwOi8vbnMuYWRvYmUuY29tL3hhcC8xLjAvc1R5c
GUvUmVzb3VyY2VFdmVudCMiCiAgICAgICAgICAgIHhtbG5zOnhtcD0iaHR0cDovL25zLmFkb2JlLmNvbS94
YXAvMS4wLyIKICAgICAgICAgICAgeG1sbnM6ZGM9Imh0dHA6Ly9wdXJsLm9yZy9kYy9lbGVtZW50cy8xLjE
vIgogICAgICAgICAgICB4bWxuczpwaG90b3Nob3A9Imh0dHA6Ly9ucy5hZG9iZS5jb20vcGhvdG9zaG9wLz
EuMC8iCiAgICAgICAgICAgIHhtbG5zOnRpZmY9Imh0dHA6Ly9ucy5hZG9iZS5jb20vdGlmZi8xLjAvIgogI
CAgICAgICAgICB4bWxuczpleGlmPSJodHRwOi8vbnMuYWRvYmUuY29tL2V4aWYvMS4wLyI+CiAgICAgICAg
IDx4bXBNTTpPcmlnaW5hbERvY3VtZW50SUQ+eG1wLmRpZDo3RTExMjdGMTE0MjA2ODExODIyQUMzQkY4QTQ
0NkZGNDwveG1wTU06T3JpZ2luYWxEb2N1bWVudElEPgogICAgICAgICA8eG1wTU06RG9jdW1lbnRJRD54bX
AuZGlkOjZBNDc2NERERTFDMzExRTU4RDkwRDBFRDc3MDNDQzc4PC94bXBNTTpEb2N1bWVudElEPgogICAgI
CAgICA8eG1wTU06SW5zdGFuY2VJRD54bXAuaWlkOjEwMWVlYzQ3LTQ1NjMtNGNkZS1hOTU4LTU5ZTVjMTcy
ZjIzNzwveG1wTU06SW5zdGFuY2VJRD4KICAgICAgICAgPHhtcE1NOkRlcml2ZWRGcm9tIHJkZjpwYXJzZVR
5cGU9IlJlc291cmNlIj4KICAgICAgICAgICAgPHN0UmVmOmluc3RhbmNlSUQ+eG1wLmlpZDo4Yjc4OGJlNi
1hOTg5LTQwZjQtOGQ5Mi1lMzUwOTNjNjYzZmI8L3N0UmVmOmluc3RhbmNlSUQ+CiAgICAgICAgICAgIDxzd
FJlZjpkb2N1bWVudElEPnhtcC5kaWQ6N0UxMTI3RjExNDIwNjgxMTgyMkFDM0JGOEE0NDZGRjQ8L3N0UmVm
```

OmRvY3VtZW50SUQ+CiAgICAgICAgIDwveG1wTU06RGVyaXZlZEZyb20+CiAgICAgICAgIDx4bXBNTTpIaXN
0b3J5PgogICAgICAgICA8cmRmOlNlcT4KICAgICAgICAgICAgPHJkZjpsaSByZGY6cGFyc2VUeX
BlPSJSZXNvdXJjZSI+CiAgICAgICAgICAgICAgIDxzdEV2dDphY3Rpb24+c2F2ZWQ8L3N0RXZ0OmFjd
Glvbj4KICAgICAgICAgICAgICAgICAgPHN0RXZ0Omluc3RhbmNlSUQ+eG1wLmlpZDoyYWFhYWQwNi1kZmU0
LTRmODItODY3Yi00MjliZTMzOTE5MzY8L3N0RXZ0Omluc3RhbmNlSUQ+CiAgICAgICAgICAgICAgIDx
zdEV2dDp3aGVuPjIwMTYtMDMtMjhUTc6MDY6MjkrMDg6MDA8L3N0RXZ0OndoZW4+CiAgICAgICAgICAgIC
AgICAgICAgIDxzdEV2dDpzb2Z0d2FyZUFnZW50PkFkb2JlIFBob3Rvc2hvcCBDQyAoTWFjaW50b3NoKTwvc3RFd
nQ6c29mdHdhcmVBZ2VudD4KICAgICAgICAgICAgICAgICAgPHN0RXZ0OmNoYW5nZWQ+Lzwvc3RFdnQ6Y2hh
bmdlZD4KICAgICAgICAgICAgICAgICAgPC9yZGY6bGk+CiAgICAgICAgICAgICAgICAgIDxyZGY6bGkgcmRmOnBhcnN
lVHlwZT0iUmVzb3VyY2UiPgogICAgICAgICAgICAgICAgICAgICA8c3RFdnQ6YWN0aW9uPnNhdmVkPC9zdEV2dD
phY3Rpb24+CiAgICAgICAgICAgICAgICAgICAgICAgICA8c3RFdnQ6aW5zdGFuY2VJRD5hZG9iZTpkaWQ6MTAxZWVjNDctN
DU2My00Y2RlLWE5NTgtNTllNWMxNzJmMjM3PC9zdEV2dDppbnN0YW5jZUlEPgogICAgICAgICAgICAgICAgICAgICA
gICA8c3RFdnQ6d2hlbj4yMDE2LTAzLTI4VDE3OjA2OjMzKzA4OjAwPC9zdEV2dDp3aGVuPgogICAgICAgICAgICAgICA
gICAgICAgICA8c3RFdnQ6c29mdHdhcmVBZ2VudD5BZG9iZSBQaG90b3Nob3AgQ0MgKE1hY2ludG9zaCk8L3
N0RXZ0OnNvZnR3YXJlQWdlbnQ+CiAgICAgICAgICAgICAgICAgICAgICAgICA8c3RFdnQ6d2hlbj4yMDE2LTAzLTI4VDE
mNoYW5nZWQ+CiAgICAgICAgICAgICAgICAgICAgICAgICA8L3JkZjpTZXE+CiAgICAgICAgICAgICAgICAgICAgICAgICA
ICAgICAgICAgICA8L3JkZjpsaT4KICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICA8L3JkZjpTZXE+CiAg
ICAgICAgICAg

IAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICA
gICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgIC
AgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgI
CAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAg
ICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICA
gICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIC
AgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgI
CAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAg
ICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICA
gICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgIC
AgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgI
CAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAg
ICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICA
KICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIC
AgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgI
CAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAg
ICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICA
gICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIC
AgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAg

ICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICA
gICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIC
AgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgI
CAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAg
ICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICA
gICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIC
AgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgI
CAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAK
ICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICA
gICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgIC
AgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgI
CAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAg
ICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICA
gICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgI
CAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAg
ICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICA
gICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIC
AgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAg

ICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICA
gICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIC
AgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKI
CAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAg
ICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICA
gICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIC
AgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgI
CAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAg
ICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICA
gICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIC
AgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgC
iAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAg
ICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICA
gICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIC
AgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgI
CAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAg
ICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICA
gICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIC
AgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgI
CAgICAgICAg

ICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICA
gICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIC
AgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgI
CAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAg
ICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICA
gICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIC
AgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgI
CAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAg
ICAgICAgICAgICAgICAgICAgICAg

```
ICAgICAgICAg
```

```
}
.btn-cart .jian{
    background:
url(
2Z0d2FyZQBBZG9iZSBJbWFnZVJlYWR5ccllPAAAA2tpVFh0WE1MOmNvbS5hZG9iZS54bXAAAAAAADw/eHBh
Y2tldCBiZWdpbj0i77u/IiBpZD0iVzVNME1wQ2VoaUh6cmVTek5UY3prYzlkIj8+IDx4OnhtcG1ldGEgeG1
sbnM6eD0iYWRvYmU6bnM6bWV0YS8iIHg6eG1wdGs9IkFkb2JlIFhNUCBDb3JlIDUuNS1jMDE0IDc5LjE1MT
Q4MSwgMjAxMy8wMy8xMy0xMjowOToxNSAgICAgICAgIj4gPHJkZjpSREYgeG1sbnM6cmRmPSJodHRwOi8vd
3d3LnczLm9yZy8xOTk5LzAyLzIyLXJkZi1zeW50YXgtbnMjIj4gPHJkZjpEZXNjcmlwdGlvbiByZGY6YWJv
dXQ9IiIgeG1sbnM6eG1wTU09Imh0dHA6Ly9ucy5hZG9iZS5jb20veGFwLzEuMC9tbS8iIHhtbG5zOnN0UmV
mPSJodHRwOi8vbnMuYWRvYmUuY29tL3hhcC8xLjAvc1R5cGUvUmVzb3VyY2VSZWYjIiB4bWxuczp4bXA9Im
h0dHA6Ly9ucy5hZG9iZS5jb20veGFwLzEuMC8iIHhtcE1NOk9yaWdpbmFsRG9jdW1lbnRJRD0ieG1wLmRpZ
Do3RTExMjdGMTE0MjA2ODExODIyQUMzQkY4QTQ0NkZGNCIgeG1wTU06RG9jdW1lbnRJRD0ieG1wLmRpZDpB
QzIyMEEzRUUxQzQxMUU1OEQ5MEQwRUQ3NzAzQ0M3OCIgeG1wTU06SW5zdGFuY2VJRD0ieG1wLmlpZDpBQzI
yMEEzREUxQzQxMUU1OEQ5MEQwRUQ3NzAzQ0M3OCIgeG1wOkNyZWF0b3JUb29sPSJBZG9iZSBQaG90b3Nob3
AgQ0MgKE1hY2ludG9zaCkiPiA8eG1wTU06RGVyaXZlZEZyb20gc3RSZWY6aW5zdGFuY2VJRD0ieG1wLmlpZ
Do4Yjc4OGJlNi1hOTg5LTQwZjQtOGQ5Mi1lMzUwOTNjNjYzZmIiIHN0UmVmOmRvY3VtZW50SUQ9InhtcC5k
aWQ6N0UxMTI3RjExNDIwNjgxMTgyMkFDM0JGOEE0NDZGRjQiLz4gPC9yZGY6RGVzY3JpcHRpb24+IDwvcmR
mOlJERj4gPC94OnhtcG1ldGE+IDw/eHBhY2tldCBlbmQ9InIiPz5eGu8TAAADkElEQVR42uxaS09TQRSeog
F8RFggCxIVaqC1YETFDaKJ/gLEqCw0xv8AMdG4xMTEpULZ4YKFC5MudEEiEV9sZKELQBCNUVzJQw3BtGrrO
eGbcqyt9L7aO40n+TLp7Z0z57szZ2bOmdmkSkw2u6xvF+EEoY0QItQTagjbCAHCCmGB8J4wQ5ggPCJ89NNH
2U24RpgkpGxiEjr2ODUm4KDuAcJVwmlCGZ59JzwnPCNMoxe+AixVAPfePkIH4ShhC/5PEu4R+givCtUjtYQ
hNM5f9wchRugiVNrQV4m6MehKQfcQ2vJUzhCWBZFBQtDl4XtbEFtGm65LOaFfjPnHhBYPP1wz2tDtDcAGV2
Q7YUT0ymWHvmfFv7mtBNoegS2OZCthHAo/w4ELLdzmImwYh022h9koFM0TwkVcGiKET7Dlod3h1y8cc78P1
ju2YUn4lCU5h4qJIg2zfw0/7VPd+VaqE1Nzrw+3a71i5NTlU2FYTM0BHxIKiCl9eKOXD2GVTni8zjiVFtiY
hM05JQbmgwZEClHYGsv1Au90f2HxDBpAKAhb2eYG/bBMvHAevx8Q3hlAiG28D5svZXthCl3YZVCAego2z2S
LNPmPVUKFQYQqYHNKu4kecidRcnAWN4hQHMGkQuifJtSG8qmBeRFtc6sk1ITytYGEtP+EJKG9KN8YSGhWTO
NpQjtRLlpQ1IcxnHIZcejOV5ZQVsuHWlm1BUU/PSAjSeUrVbJOmSox0YS+odxhoe4NbBDdFt7O3LTYQywr8
uEcuu2ggZ3SCtvnZA+9RdloIKFGsbdLE9JTX9hAQmG5HmlCEyg7DCR0DOXLktycmh4+dMLm2WwBnk44XDCI
0EWUd7P9WW9YCN6QLQTPlSSJGkBoYKMkCcthtZ7GavYxmYjKM42lfYmZjyn/JhrHVJ6JRhaZCu7xISHLqWC
WbrWerG/3EZl2ZSNZn+l0HPT55Thlwcmk1XngFSryJDAPW0aVg/PWzCPJYgw/144ktchD4wScslCHxj3K5U
NjOfyiIuYf83idioipWftMuRcNyYsXCTTU4KJ+3vXfEr3CbZ31eijwdZU76u+rMbzztXs1phM6NJEk2qi1M
06dxPJX1J+Xl1bhuE/UWhaWt/VfMpIwnCprQqR5HBONdnR9eel6ZsBWSNHXy6aU/TzctPLB9bJc5PgU4Ah6
gUOSGjE7yQuA3Hsv1NoFwA/qv2SX3wIMABYdQ1WkGFvcAAAAAElFTkSuQmCC) no-repeat;
    background-size: 100% 100%;
}
.btn-cart.incart .count,
.btn-cart.incart a.jian{
    display: table-cell;
}

.btn-cart a.jia:active,
.btn-cart a.jian:active{
    background: #eb3c3c;
```

```
    color:#fff;
    background:
url(
2Z0d2FyZQBBZG9iZSBJbWFnZVJlYWR5ccllPAAAA2tpVFh0WE1MOmNvbS5hZG9iZS54bXAAAAAAADw/eHBh
Y2tldCBiZWdpbj0i77u/IiBpZD0iVzVNME1wQ2VoaUh6cmVTek5UY3prYzlkIj8+IDx4OnhtcG1ldGEgeG1
sbnM6eD0iYWRvYmU6bnM6bWV0YS8iIHg6eG1wdGs9IkFkb2JlIFhNUCBDb3JlIDUuNS1jMDE0IDc5LjE1MT
Q4MSwgMjAxMy8wMy8xMy0xMjowOToxNSAgICAgICAgICAgIj4gPHJkZjpSREYgeG1sbnM6cmRmPSJodHRwOi8vd
3d3LnczLm9yZy8xOTk5LzAyLzIyLXJkZi1zeW50YXgtbnMjIj4gPHJkZjpEZXNjcmlwdGlvbiByZGY6YWJv
dXQ9IiIgeG1sbnM6eG1wTU09Imh0dHA6Ly9ucy5hZG9iZS5jb20veGFwLzEuMC9tbS8iIHhtbG5zOnN0UmV
mPSJodHRwOi8vbnMuYWRvYmUuY29tL3hhcC8xLjAvc1R5cGUvUmVzb3VyY2VSZWYjIiB4bWxuczp4bXA9Im
h0dHA6Ly9ucy5hZG9iZS5jb20veGFwLzEuMC8iIHhtcE1NOk9yaWdpbmFsRG9jdW1lbnRJRD0ieG1wLmRpZ
Do3RTExMjdGMTE0MjA2ODExODIyQUMzQkY4QTQ0NkZGNCIgeG1wTU06RG9jdW1lbnRJRD0ieG1wLmRpZDow
NzMzMTcyN0UxQ0MxMUU1OEQ5MEQwRUQ3NzAzQ0M3OCIgeG1wTU06SW5zdGFuY2VJRD0ieG1wLmlpZDowNzM
zMTcyNkUxQ0MxMUU1OEQ5MEQwRUQ3NzAzQ0M3OCIgeG1wOkNyZWF0b3JUb29sPSJBZG9iZSBQaG90b3Nob3
AgQ0MgKE1hY2ludG9zaCkiPiA8eG1wTU06RGVyaXZlZEZyb20gc3RSZWY6aW5zdGFuY2VJRD0ieG1wLmlpZ
Do4Yjc4OGJlNi1hOTg5LTQwZjQtOGQ5Mi1lMzUwOTNjNjYzZmIiIHN0UmVmOmRvY3VtZW50SUQ9InhtcC5k
aWQ6N0UxMTI3RjExNDIwNjgxMTgyMkFDM0JGOEE0NDZGRjQiLz4gPC9yZGY6RGVzY3JpcHRpb24+IDwvcmR
mOlJERj4gPC94OnhtcG1ldGE+IDw/eHBhY2tldCBlbmQ9InIiPz71ecqTAAADRUlEQVR42tyaPWgTYRjH36
SlhHpKh+ggqPWDFl2UUAehHVqhU9NFYloViptddGg7CR0Up1LQxeoiJaBoxSXp5GAKHTpYUBc1flYFh9qhS
Cgh0Oj/uXtOj3Dp5S7v5T7+8Iek0Pd9frn343mf9yK/entFmNQqub0DcD/cA3fDnXAc3gVH4CK8Aa/BBXgV
zsM/ZAUQkfCEDsJj8Ah8wmEbb+HHcAb+5hXQSfg6fA6OSvqBK/Az+Bb8xkkDTgLZB8/Dr+CURBg9nhS3Pc9
9uQqU4rE/xnPCLUW4jwL3KR2oDb4LL8AdTVy0OrjPOY5BCpAC5+BxD1fjKxyD0ihQO/wcHvTBFjPIsbQ7BW
rjX+WMj/ZNiiW70/DbCeg2PODDZOAsfMcuUNrjOVPPnBqpF2g/fC8Aadscx2oJNNPkpbmRJX3GCigBj8rsV
ZmcFPGlJRHP54UyNSUbapRjrgk0LTsDiCWTQrS0IK9vFbGhITcyiulaQIfgpPSBEY2af5YnivmwGdAlyYlm
s0QxXzYDuhjgg2q6GohOmscDDNQFHzECDYjgq98I1BMCoFNGoK4QAHUbgY6GAEidQ3oZa6/tDGBiQsSGh23
vLfHlZYsySUWUsllRnJ11kgr9e0J7HGUAbmyUaFNt2752OymSBGKXJf22+4+lXE4dHtJFQ47atq+imtxxof
GTWwtD9ZzZ6Otz6+F8ho9FDV+Cri/GIfchBEAFI9BqCIBeG4FehAAobwSi+5l3AYb5WD2HSA8DDPTE7IBHQ
JUAwlDMD8yA1oRW+pW+UZp+lieK+asZEOkm/Edmb+quv72turS4KBuGYr1h/IPZlSQNvQsBGW6PqmshZskp
VQM3AwCzybEKK6Cfwt+Fel3jHKslEImu2P1csL/PMYp6gUjXfJpBUExXrc5DZioLrcy64iOYFY6p7ASItCX
+3216Lf2ud6ueE6vVSTDJ49bLOZPUT6WNAunDj64Bzzd5Sae+0tx32U5NoV49FVpBLyM7ozDJADLc14KTIo
kdrQvttZUEA8pM0CrcZoL7WLfbgMzXy+h60OkNxntOYzx9vawWHN0CnBZavbxTaC8AKoYFRn8BkOoYL/mk+
V1WAJGwvaL5V4ABAG0LrGiYgMe/AAAAAElFTkSuQmCC) no-repeat;
    background-size: 100% 100%;
}
.btn-cart a.jian:active{
    background:
url(
2Z0d2FyZQBBZG9iZSBJbWFnZVJlYWR5ccllPAAAA2tpVFh0WE1MOmNvbS5hZG9iZS54bXAAAAAAADw/eHBh
Y2tldCBiZWdpbj0i77u/IiBpZD0iVzVNME1wQ2VoaUh6cmVTek5UY3prYzlkIj8+IDx4OnhtcG1ldGEgeG1
sbnM6eD0iYWRvYmU6bnM6bWV0YS8iIHg6eG1wdGs9IkFkb2JlIFhNUCBDb3JlIDUuNS1jMDE0IDc5LjE1MT
Q4MSwgMjAxMy8wMy8xMy0xMjowOToxNSAgICAgICAgICAgIj4gPHJkZjpSREYgeG1sbnM6cmRmPSJodHRwOi8vd
3d3LnczLm9yZy8xOTk5LzAyLzIyLXJkZi1zeW50YXgtbnMjIj4gPHJkZjpEZXNjcmlwdGlvbiByZGY6YWJv
dXQ9IiIgeG1sbnM6eG1wTU09Imh0dHA6Ly9ucy5hZG9iZS5jb20veGFwLzEuMC9tbS8iIHhtbG5zOnN0UmV
mPSJodHRwOi8vbnMuYWRvYmUuY29tL3hhcC8xLjAvc1R5cGUvUmVzb3VyY2VSZWYjIiB4bWxuczp4bXA9Im
```

```
h0dHA6Ly9ucy5hZG9iZS5jb20veGFwLzEuMC8iIHhtcE1NOk9yaWdpbmFsRG9jdW1lbnRJRD0ieG1wLmRpZ
Do3RTExMjdGMTE0MjA2ODExODIyQUMzQkY4QTQ0NkZGNCIgeG1wTU06RG9jdW1lbnRJRD0ieG1wLmRpZDpB
QzIyMEE0MkUxQzQxMUU1OEQ5MEQwRUQ3NzAzQ0M3OCIgeG1wTU06SW5zdGFuY2VJRD0ieG1wLmlpZDpBQzI
yMEE0MUUxQzQxMUU1OEQ5MEQwRUQ3NzAzQ0M3OCIgeG1wOkNyZWF0b3JUb29sPSJBZG9iZSBQaG90b3Nob3
AgQ0MgKE1hY2ludG9zaCkiPiA8eG1wTU06RGVyaXZlZEZyb20gc3RSZWY6aW5zdGFuY2VJRD0ieG1wLmlpZ
Do4Yjc4OGJlNi1hOTg5LTQwZjQtOGQ5Mi1lMzUwOTNjNjYzZmIiIHN0UmVmOmRvY3VtZW50SUQ9InhtcC5k
aWQ6N0UxMTI3RjExNDIwNjgxMTgyMkFDM0JGOEE0NDZGRjQiLz4gPC9yZGY6RGVzY3JpcHRpb24+IDwvcmR
mOlJERj4gPC94OnhtcG1ldGE+IDw/eHBhY2tldCBlbmQ9InIiPz4ibqSTAAADAklEQVR42tyaPWzTQBTHzx
XtEALqAAxIQPlQK1hAURkqdWmRumVCIamCVLFUdIGJCakDiKkLLBQ6VFWlQChigY2Bjh0oXwsQPgtIDIWhQ
lWGtEr5P/sZrMqJ7cs59vlJvyFD7t7POV/efRi/BgdFkmKH4vYOgCHQD/pAD9gDdgIDrIPfYAVUwDJYBD9U
JWAo+IUOgjFQACck23gLymAefItK6CS4Cs6CDkUPuA4egRvgjUwDMonsA3PgFcgplLHzyXHbc9xXqEI5Hvt
j/E6EFQb3UeE+lQt1gdtgAXS3cdLq5j6nOQclQmnwBExEOBtf5BzSrQqlwFMwEoO/mBHOJSUr1MVPZSBG/5
uUy+Nmw6+Z0E0wHMNi4Ay4FVQoH/E74+edKvgV2g/uaFC2TXOunkJTbZ6aW5nSp7yEMmBUo+J6lHNuKDQZc
gUQRkUx2UjoEMhquASinA+7CZ1XXGi2KyjnC25CRY0XqvntQrTSPK6xUC844hQaFvrHkFOoPwFCp5xCvQkQ
6nMKHU2AkPkO2dtYe4N+OzU+LlIF1IednWrT2tgQ1XJZVGdmZEqhf7/Q7sBCxaJ6GQq0aT6o4LFLZpNEi39
Zij9Bv1gtlczhoTw2N80hJxHrZnHHG42fEjAxfAbHOhwfdI8vziH3IQFCFafQcgKEXjuFniVAaNEpROcz7z
SW+bj9HaIoaSz0wG2BR0J1DWUo51k3oRVhbf3qFpTzVzchiutgSyMZyvWaW+ljxwtwXyMhyvVlMyGKK2BNA
5k1zlV4Cf0U8d6ot2OCc/UUoqByN84b9nc5R+FXiOJyTCsIyumS13rILWrC2mZdipHMEudUkxEy13Hi/9lm
1GGf9Vb9rFi9VoJZHrdRvjNZe1XaqpA9/OgY8Fybp3TqK89914LsKfiNh8La0JsPuaLY4j6orwWZTZIgsSq
saysZFlRZ0Na5zQz3sRq0AZXXy+h4UPYE4z24JyK+XtZIjk4BTgtrv7xHWBcA044Jxr4ASPsYz3ml+V1VAk
bSrmj+FWAAhXKURfU1tf0AAAAASUVORK5CYII=) no-repeat;
    background-size: 100% 100%;
}

/* 菜品分类**/
.meal-type-box{
    margin: 0.94rem 0.4rem 0 0.47rem;
}
.meal-type {
    display: table;
    width: 100%;
    border: 1px solid #eb3c3c;

}
.meal-type .title,
.meal-type .desc{
    display: table-cell;
    height: 2.5rem;
    line-height: 2.5rem;
    color: #fff;
    font-size: 1.1rem;
    text-align: center;
}
.meal-type .title{
```

```
    width: 30%;
    font-weight: bold;
}
.meal-type .desc{
    color: #a3a3a3;
    letter-spacing:3px;
}
/* 新菜 */
.new-meal{
    color: #eb3c3c;
}
.new-meal .title{
    background: #eb3c3c;
}
.new-meal .desc{
    color: #eb3c3c;
}

/* 焦耳鲜辣*/
.jexl{
    border: 1px solid #1e9664;
}
.jexl,
.jexl .desc{
    color: #1e9664;
}
.jexl .title{
    background: #1e9664;
}
/* 焦耳麻辣*/
.jemj{
    border: 1px solid #aa3737;
}

.jemj,
.jemj .desc{
    color: #aa3737;
}
.jemj .title{
    background: #aa3737;
}
/* 焦耳双拼*/

.jesp{
```

```
    border: 1px solid #5f6464;
}

.jesp,
.jesp .desc{
    color: #5f6464;
}
.jesp .title{
    background: #5f6464;
}

/* 饮料*/
.yl{
    border: 1px solid #00a0f0;
}

.yl,
.yl .desc{
    color: #00a0f0;
}
.yl .title{
    background: #00a0f0;
}


/* 购物车 */
footer{
    background: hsla(0,0%,100%,.95);
    position: fixed;
    bottom: 0;
    width: 100%;
    min-height: 3.5rem;
    padding: 0 10px;
    z-index: 998;
    border: 1px solid rgba(0,0,0,0.1);
}

footer::before {
    content: '';
    position: absolute;
    top: 0; right: 0; bottom: 0; left: 0;
    -webkit-filter: blur(3.5rem);
    filter: blur(3.5rem);
    z-index: -1;
```

```
}
footer .total{
    letter-spacing:2px;
}
footer .total.able{
    color: #eb3c3c;
}
.bottom{
    padding: 0 0 0 3.5rem;
    height: 3.5rem;

}
.bottom .table{
    height: 3.5rem;
}
a.btn-toPay{
    display: inline-block;
    width: 6rem;
    padding: 0.4rem 0;
    border: 1px solid #969696;
    color: #969696;
    border-radius: 2px;
    text-align: center;
    font-size: 1.2rem;
}
a.btn-toPay.able{
    background: #eb3c3c;
    color: #fff;
    border: none;
}
.icon-cart-box{
    float: left;
    width:3.5rem;
    height: 3.5rem;
    position: relative;
}
.cart-num {
    border: 1px solid #eb3c3c;
    height: 1.75rem;
    min-width: 1.75rem;
    width: auto;
    position: absolute;
    border-radius: 1.75rem;
    text-align: center;
```

```
    top: -2.55rem;
    line-height: 1.75rem;
    left: 1.75rem;
    margin-left: -0.87rem;
    font-size: 0.87rem;
    color: #eb3c3c;
    font-size: 0.85rem;
    font-weight: bold;
    background: #fff;
}

.icon-cart{
    background: hsla(0,0%,100%,.95);
    position: absolute;
    /* border-radius: 60px; */
    border-top-left-radius: 200px;
    border-top-right-radius: 200px;
    border-bottom-left-radius: 200px;
    border-bottom-right-radius: 200px;
    bottom: 0.8rem;
    width: 3.5rem;
    height: 3.5rem;
    background: url(../../img/cart.png) #fff no-repeat;
    background-size: 100% auto;
}

.cartItems{
    position: relative;
    min-height: 0;
    /*height: 0;*/
    max-height: 14rem;
    /*height: 14rem;*/
    overflow: hidden;
}
.cartItems  .cell{
    padding: 0.625rem 0;
    border-bottom: 1px solid #c9c9c9;
    font-size: 1rem;
}
.cartItems div.row:first-child .cell{
    padding: 0 0 0.625rem 0;
}
.btn-cart-pop{
    display: block;
```

```
    position: absolute;
    width: 100%;
    height: 100%;
    top: 0;
    left: 0;
    z-index: 998;
}
```

- js/templates/home.html

```
<link rel="stylesheet" href="css/templates/home.css">

<!-- 头部 #start -->
<header>
 <div class="addr">
  <span class="icon icon-addr"></span>
  <a href="locationAddr.html" id="address">尚硅谷</a>
 </div>
 <div class="user">
  <span class="icon icon-user"></span>
  <a href="personalCenter.html">个人中心</a>
 </div>
</header>
<!-- 头部 #end -->

<!-- 轮播 #start -->
<div class="banner">
 <div class="swiper-container" id="bannerSwiper">
  <div class="swiper-wrapper" id="slides">
   <div class="swiper-slide">
    <a href="#">
     <img src="img/bar1.jpg"/>
    </a>
   </div>
   <div class="swiper-slide">
    <a href="#">
     <img src="img/bar2.png"/>
    </a>
   </div>
  </div>
 </div>
 <!-- Add Pagination -->
 <div class="swiper-pagination"></div>
</div>
```

```
<!-- 轮播 #end -->

<!-- 菜品列表 #start -->
<div class="wrap meals">
 <div id="info">
  <ul>
   <li >
    <div class="meal-item">
     <a class="img meal-img" href="javascript:">
      <img src="img/default.jpg" />
     </a>
     <div class="meal-name">熏鸡肉沙拉</div>
     <div class="sales">月售 211 份</div>
     <div class="buy">
      <div class="price">
       &yen;
       <span class="ml2 f150 fb">23</span>
        <span>
         <s>&yen;56</s>
        </span>
      </div>
      <div class="option-cart" >
       <div class="btn-cart in fr">
        <a class="jian"></a>
        <span class="count">0</span>
        <a class="jia"></a>
       </div>
      </div>
     </div>
    </div>
   </li>
   <li >
    <div class="meal-item">
     <a class="img meal-img" href="javascript:">
      <img src="img/default.jpg" />
     </a>
     <div class="meal-name">熏鸡肉沙拉</div>
     <div class="sales">月售 211 份</div>
     <div class="buy">
      <div class="price">
       &yen;
       <span class="ml2 f150 fb">23</span>
        <span >
         <s>&yen;56</s>
```

```
        </span>
      </div>
      <div class="option-cart">
       <div
         class="btn-cart in fr">
        <a class="jian"></a>
        <span class="count">0</span>
        <a class="jia"></a>
       </div>
      </div>
     </div>
    </div>
   </li>
   <div class="clearfix"></div>
  </ul>
 </div>
</div>
<!-- 菜品列表 #end -->

<!-- 购物车 #start -->
<footer>
 <div class="blurrycontent"></div>
 <div id="cart" style="position:relative">
  <style>
   .swiper-container {
    width: 100%;
    height: 100%;
   }
   .swiper-slide {
    height: auto;
    -webkit-box-sizing: border-box;
    box-sizing: border-box;
   }
  </style>
  <!-- 购物车图标-->
  <div class="icon-cart-box" >
   <div class="cart-num" id="total_count">1</div>
   <div class="icon-cart">
    <a class="btn-cart-pop">
    </a>
   </div>
  </div>
  <div class="table w100 cartItems" id="cartItems" >
   <div class="swiper-container" id="cartSwiper">
```

```
    <div class="swiper-wrapper">
     <div class="swiper-slide">
      <div class="table w100">
       <div class="row" >
        <div class="cell">熏鸡肉沙拉</div>
        <div class="cell">
         &yen;<span class="ml2 f120 fb">23</span>
        </div>
        <div class="cell tr">
         <div class="option-cart fr">
          <div class="btn-cart in fr">
           <a class="jian cart-num-btn"></a>
           <span class="count ">1</span>
           <a class="jia cart-num-btn"></a>
          </div>
         </div>
        </div>
       </div>
      </div>
     </div>
    </div>
   </div>
  </div>
  <!-- Add Scroll Bar -->
  <div class="swiper-scrollbar"></div>

  <div class="bottom">
   <div class="table w100">
    <div class="cell total tc fb f120" >
     共计：&yen;<span id="total_price">0</span>元</div>
    <div class="cell tr">
     <a href="orderConfirm.html" class="btn-toPay">选好了</a>
    </div>
   </div>
  </div>
 </div>
</footer>
<!-- 购物车 #end -->
```

3) 定义相关模块 js

- js/app/app.js

```
/**
 * angular 应用module 模块
 * 1. 指定应用模块名(启动时需对应)
 * 2. 引入用于路由和验证的2 个angular 模块
 */
define(['angular'], function (angular) {
    return angular.module('jouleApp', ['ngRoute', 'ngMessages'])
})
```

- js/controllers/XxxCtrl.js

```
/**
 * 首页Controller 模块
 */
define(['app'], function (app) {
  return app.controller('HomeCtrl', ['$scope', '$rootScope',
    function ($scope, $rootScope) {
      $rootScope.appTitle = '首页';

    }]);
});
```

- js/routes/appRoute.js

```
/**
 * 前端路由配置模块
 */
define(['app'], function (app) {
    return app.config(['$routeProvider', function ($routeProvider) {
        //注册路由
        $routeProvider
            .when('/home', {
                templateUrl : 'js/templates/home.html',
                controller : 'HomeCtrl'
            })
            .when('/personal', {
                templateUrl : 'js/templates/personalCenter.html',
                controller : 'PersonalCtrl'
            })
            .when('/add_new_addr', {
                templateUrl : 'js/templates/addNewAddr.html',
                controller : 'AddNewAddrCtrl'
            })
            .when('/addr_manage', {
                templateUrl : 'js/templates/addrManage.html',
                controller : 'AddrManageCtrl'
```

```
            })
            .when('/choose_oordinate', {
                templateUrl : 'js/templates/chooseCoordinate.html',
                controller : 'ChooseCoordinateCtrl'
            })
            .when('/feedback', {
                templateUrl : 'js/templates/feedback.html',
                controller : 'FeedbackCtrl'
            })
            .when('/location_addr', {
                templateUrl : 'js/templates/locationAddr.html',
                controller : 'LocationAddrCtrl'
            })
            .when('/login', {
                templateUrl : 'js/templates/login.html',
                controller : 'LoginCtrl'
            })
            .when('/order_confirm', {
                templateUrl : 'js/templates/orderConfirm.html',
                controller : 'OrderConfirmCtrl'
            })
            .otherwise('/home')
    }]);
});
```

4) 定义入口 js

- js/main.js

```
/**
 * 入口主 js
 * 1. 配置 requirejs
 * 2. 加载所有模块, 启动应用
 */

// 1. 配置 requirejs
//配置 require
require.config({
  // 基本路径(简化 paths 编写)
  baseUrl : 'js/',
  //映射模块: name:path
  paths : {
    // lib 模块
```

```
    'angular' : 'libs/angular',
    'angularMessages' : 'libs/angularMessages',// 注意修改库文件名
    'angularRoute' : 'libs/angularRoute', // 注意修改库文件名

    // 应用模块
    'app' : 'app/app',

    // controllers
    'HomeCtrl' : 'controllers/HomeCtrl',
    'PersonalCtrl' : 'controllers/PersonalCtrl',

    // routes
    'appRoute' : 'routes/appRoute',
  },
  // 非AMD 模块
  shim : {
    'angular' : {
      exports : 'angular'
    },
    'angularMessages' : {
      exports : 'angularMessages',
      deps : ['angular']
    },
    'angularRoute' : {
      exports : 'angularRoute',
      deps : ['angular']
    }
  }
});

// 2. 加载所有模块, 启动应用
require(
  // 模块标识名的数组
  ['angular', 'angularMessages', 'angularRoute', 'app',
  'HomeCtrl', 'PersonalCtrl', 'appRoute'],
  // 当模块加载完后调用
  function (angular) {
    // 启动当前 angular 应用
    angular.bootstrap(document, ['jouleApp'])
  }
);
```

5) 完成主 HTML 页面

- index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1,
user-scalable=no">
  <!--动态显示标题-->
  <title ng-bind="appTitle"></title>
  <!--引入全局 css-->
  <link rel="stylesheet" href="../css/reset.css">
  <link rel="stylesheet" href="../css/common.css">
  <link rel="stylesheet" href="../css/layout.css">
</head>
<body>
<ng-view></ng-view>
<!--引入 requirejs, 加载入口 main.js-->
<script type="text/javascript" src="js/libs/require.min.js"
data-main="js/main"></script>
</body>
</html>
```

启动服务器访问测试:

6) 实现主页面与各个页面间的路由跳转

- 完成各个页面对应的 Controller 的定义
- 在 main.js 中配置各个 Controller 的标识名称， 并加载所有新的 Controller
- 将页面的跳转路径改为路由路径:

```
<a href="#/personal">个人中心</a>
```

2.4. 实现登陆界面功能

1) 表单验证

```
<link rel="stylesheet" href="css/templates/login.css">

<form name="loginForm">
  <div class="lg_form lg_form1">
    <span class="lg_tel">手机号</span>
    <input type="text" class="lg_phone" placeholder="请输入您的手机号"
           autocomplete="off" name="phone" ng-required="true"
           ng-pattern="/^1[0-9]{10}$/" ng-model="user.phone"/>
    <div ng-messages="loginForm.phone.$error"
         ng-show="loginForm.phone.$dirty&&loginForm.phone.$invalid"
         ng-class="'errorMsg'">
      <span ng-message="required" ng-bind="'手机号必须指定'"></span>
      <span ng-message="pattern" ng-bind="'必须是手机号格式'"></span>
      <span></span>
    </div>
  </div>

  <div class="lg_form lg_form2">
    <span class="fl">验证码</span>
    <input type="text" class="lg_checkCode" placeholder="请输入短信验证码"
           autocomplete="off" name="code" ng-required="true"
           ng-pattern="/^[0-9]{6}$/" ng-model="user.code" ng-disabled="!timing"/>
    <button class="lg_btn btn" ng-click="startTime()"
            ng-bind="btnText" ng-disabled="timing"></button>
    <div ng-messages="loginForm.code.$error"
         ng-show="loginForm.code.$dirty&&loginForm.code.$invalid"
         ng-class="'errorMsg'">
```

```
      <span ng-message="required" ng-bind="'验证码必须指定'"></span>
      <span ng-message="pattern" ng-bind="'必须是 6 位'"></span>
      <span></span>
    </div>
  </div>

  <button class="lp_loginBtn btn" ng-disabled="loginForm.$invalid"
          ng-click="login()">登录
  </button>
</form>
```

2) 验证码界面效果

```
define(['app'], function(app, storageUtil) {
  return app.controller('LoginCtrl', ['$scope', '$rootScope', '$interval',
    function($scope, $rootScope, $interval) {
      $rootScope.appTitle = '登 陆'

      $scope.timing = false;//是否正在计时
      $scope.btnText = '获取验证码';

      //开始计时
      $scope.startTime = function() {
        //检查手机号是否合法，如果不合法提示，结束
        if ($scope.loginForm.phone.$invalid) {
          $scope.loginForm.phone.$dirty = true;
          return;
        }
        var time = 60;
        //time = 16; //测试
        $scope.btnText = time + 's 重新获取';
        $scope.timing = true;//开始计时
        //使用$interval 启动定时器
        var stop = $interval(function() {
          //统计剩余时间
          time--;
          //不断改变 btnText
          $scope.btnText = time + 's 重新获取';
          //当计时完成，停止计时，更新 btnText
          if (time === 0) {
            $interval.cancel(stop);
            $scope.btnText = '获取验证码';
```

```
            $scope.timing = false;//停止计时
          }
        }, 1000)
      }
    }])
})
```

3) 与服务器 ajax 交互的 service 模块: serverService

- 后台暂时借用 dc_grunt 项目(app.js)

```
//在 app.use('/', routes);之前
app.use(function(req, res, next) {
  res.set('Access-Control-Allow-Origin', '*'); // cors 解决跨域问题
  next();
});
```

- js/services/serverService.js

```
/**
 * 与服务器 ajax 交互的 service 模块
 */
define(['app'], function(app) {
  const BASE_URL = 'http://localhost:4000';
  return app.factory('serverService', ['$http', '$q',
    function($http, $q) {
      /*
       发送验证码
       */
      function sendCode (phone) {

        var defer = $q.defer();

        $http.get(BASE_URL + '/sendcode?phone=' + phone)
          .success(function(result) {
            defer.resolve(result)
          })
        return defer.promise;
      }

      /*
       登陆
       */
      function login (user) {
```

```
        var defer = $q.defer();
        $http({
          url: BASE_URL + '/login',
          method: 'POST',
          data : 'phone='+user.phone+'&code='+user.code,
          //data: user  //如果不是跨域请求，才可以
          headers :  {'Content-Type': 'application/x-www-form-urlencoded'}
        })
          .success(function(result) {
            defer.resolve(result);
          })
          .error(function() {
            defer.reject();
          })
        return defer.promise;
      }

      /*
       提交意见反馈
       */
      function feedback (feedback) {
        var defer = $q.defer();
        $http({
          url: BASE_URL + '/feedback',
          method: 'GET',
          params: { data: feedback },
        })
          .success(function(result) {
            defer.resolve(result);
          })
        return defer.promise;
      }

      return {
        sendCode: sendCode,
        login: login,
        feedback: feedback
      }
    }]);
});
```

4) ajax 请求发送验证码

```
// 向服务器发送ajax 请求: 生成随机验证码, 并发给指定的号码
serverService.sendCode($scope.user.phone)
  .then(function (result) {
    console.log(result);
  });
```

5) 浏览器端数据存储的工具模块: storageUtil

```
/**
 * 浏览器端数据存储的工具模块
 * sessionStorage 存储
 * localStorage 存储
 * 形式: key=value
 *
 */
define(function() {

  return {
    KEYS: {   //存储所有的key
      USER: '_user_',   //用户
      EDIT_ADDR: '_edit_addr_', //修改的地址
      MAP_ADDR: '_map_addr_', //地图定位的地址
      INPUT_ADDR: '_input_addr_', //用户输入的地址
      CART: '_cart_', //购物车信息
      ORDER_ADDR: '_order_addr_' //下单的地址
    },

    session: {
      //保存
      setItem: function(key, value) {
        //如果value 是对象, 转换成对应的json
        if (value instanceof Object) {
          value = JSON.stringify(value);
        }
        //保存
        sessionStorage.setItem(key, value)
      },
      //读取
      getItem: function(key) {
        var value = sessionStorage.getItem(key);
```

```
        //如果value是json对象/json数组，解析成js对象/数组
        if (value != null && (value.indexOf('{') === 0 || value.indexOf('[') === 0)) {
          value = JSON.parse(value);
        }
        return value;
      },
      //删除
      removeItem: function(key) {
        sessionStorage.removeItem(key);
      }
    },

    local: {
      //保存
      setItem: function(key, value) {
        //如果value是对象，转换成对应的json
        if (value instanceof Object) {
          value = JSON.stringify(value);
        }
        //保存
        localStorage.setItem(key, value)
      },
      //读取
      getItem: function(key) {
        var value = localStorage.getItem(key);
        //如果value是json对象/json数组，解析成js对象/数组
        if (value != null && (value.indexOf('{') === 0 || value.indexOf('[') === 0)) {
          value = JSON.parse(value);
        }
        return value;
      },
      //删除
      removeItem: function(key) {
        localStorage.removeItem(key);
      }
    }
  }
})
```

6) ajax 请求登陆

```
//登陆
```

```
$scope.login = function () {
  serverService.login($scope.user)
    .then(function (result) {
      //判断登陆是否成功
      if(result.code===1) {
        alert('验证码不正确!');
      } else {
        var user = result.data;
        alert(user.phone+'登陆成功');
        //保存user到local
        //localStorage.setItem('_user_', JSON.stringify(user));
        storageUtil.local.setItem(storageUtil.KEYS.USER, user);
        //跳转到home
        window.location = '#/home'
      }
    }, function () {
      alert('登陆出错啦');
    })
}
```

2.5. 个人中心界面功能

1) personalCenter.html

```
<link rel="stylesheet" href="css/templates/personalCenter.css">

<div class="pc_main">
  <div class="pc_banner">
    <img src="img/pc_banner.png"/>
    <p class="pc_mobielPhoe">您好，{{user.phone}}</p>
  </div>
  <div class="pc_list">
    <ul>
      <li class="pc_orderList" ng-click="showOrders()">
        <i class="pc_myorder"></i>
        <span>我的订单</span>
      </li>
      <li class="pc_addr" ng-click="toAddrsUI()">
        <i></i>
        <span>地址管理</span>
      </li>
      <li class="pc_feedBack" ng-click="toFeedBackUI()">
```

```
        <i class="pc_opinion"></i>
        <span>意见反馈</span>
      </li>
    </ul>
  </div>
</div>
```

2) PersonalCtrl.js

```
define(['app', 'storageUtil'], function(app, storageUtil) {
  return app.controller('PersonalCtrl', ['$scope', '$rootScope',
    function($scope, $rootScope) {
      $rootScope.appTitle = '个人中心'

      //检查用户是否已经登陆, 如果没有提示跳转到login
      var user = storageUtil.local.getItem(storageUtil.KEYS.USER);
      if (user == null) {
        alert('请先登陆');
        window.location = '#/login';
        return;
      }

      $scope.user = user;

      $scope.toAddrsUI = function() {
        window.location = '#/addr_manage'
      }

      $scope.toFeedBackUI = function() {
        window.location = '#/feedback'
      }

      $scope.showOrders = function() {
        // window.location = '/order/orderList?userId=' + user._id;
      }
    }])
})
```

2.6. 意见反馈界面功能

1) feedback.html

```
<link rel="stylesheet" href="css/templates/feedback.css">

<nav class="nav_com">
  <div class="nv_main">描述</div>
</nav>
<section class="section_com">
  <div class="st_com">
  <textarea class="fb_val" placeholder="请尽情说出你的不愉快，我们一定改进"
            ng-model="feedback.content"></textarea>
  </div>
</section>
<footer class="footer_com">
  <button class="fb_btn btn" ng-click="submit()">提交</button>
</footer>
```

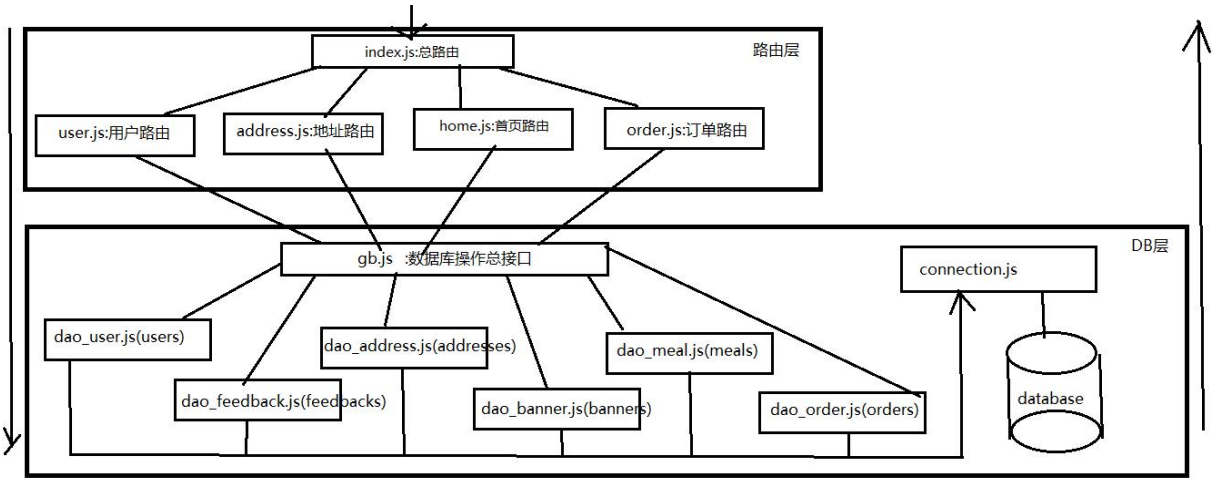
2) FeedbackCtrl.js

```
define(['app', 'storageUtil'], function(app, storageUtil) {
  return app.controller('FeedbackCtrl', ['$scope', '$rootScope', 'serverService',
    function($scope, $rootScope, serverService) {
      $rootScope.appTitle = '意见反馈';

      //读取user
      var user = storageUtil.local.getItem(storageUtil.KEYS.USER);
      $scope.feedback = {
        user_id: user._id,
        phone: user.phone
      };

      //提交ajax 请求
      $scope.submit = function() {
        serverService.feedback($scope.feedback)
          .then(function(result) {
            alert('吐槽成功!');
            window.location = '#/home'
          });
      };
```

```
    }]);
});
```

2.7. 搭建自己的后台

1) 设计后台整体结构



2) 实现 connection.js 的编码和测试

- 安装依赖包: npm install mongoose --save
- Mongoose 官方文档: http://www.nodeclass.com/api/mongoose.html

```
/**
 * 数据库操作的工具类
 * 连接数据库
 * 断开数据库连接
 */
var mongoose = require('mongoose');
exports.mongoose = mongoose;
//console.log(mongoose);

/*
 连接数据库
 */
function connect () {
  //连接
  mongoose.connect('mongodb://127.0.0.1/atguigu_o2o2');
  //获取连接
```

```
  var connection = mongoose.connection;
  //绑定连接错误监听
  connection.on('error', console.error.bind(console, 'connection error:'));
  //绑定连接成功监听
  connection.once('open', function(callback) {
    console.log('WE ARE CONNECTED!');
  });
}
exports.connect = connect;
//connect();

/*
 断开连接
 */
var disconnect = function() {
  mongoose.disconnect();
}
exports.disconnect = disconnect;
//disconnect();
```

3) 实现 dao 的编码和测试

- dao_user.js

```
/**
 * 操作 users 集合的 dao 模块
 */
var connection = require('./connection');

var mongoose = connection.mongoose;

//test #start
/*connection.connect();
 function callback(error, user) {
 console.log(error, user);
 }*/
//test #end

//schema
/*
    //构造函数
var Schema = mongoose.Schema;
    //实例对象
```

```
var userSchema = new Schema({
    phone : String
})
*/
var userSchema = new mongoose.Schema({
    phone : String
});

//model
var UserModel = mongoose.model('user', userSchema); //find/update/delete

//CRUD 函数
function getUser(phone, callback) {
    UserModel.findOne({phone:phone}, callback);
}
exports.getUser = getUser;
//getUser('13100000000', callback);



function addUser(phone, callback) {
    var userModel = new UserModel({phone:phone});
    userModel.save(callback)
}
exports.addUser = addUser;
//addUser('12312332112', callback)
```

- dao_feedback.js

```
/**
 * 操作feedbacks 集合的dao 模块
 */
var connection = require('./connection')

//test #start
/*connection.connect();
 function callback(error, feedback) {
 console.log(error, feedback);
 }*/
//test #end

//mongoose
var mongoose = connection.mongoose;

//schema
var feedbackSchema = new mongoose.Schema({
```

```
  user_id: String,
  phone: String,
  content: String,
  create_time: Date
});

//model
var FeedbackModel = mongoose.model('feedback', feedbackSchema);

//addFeedback()
function addFeedback (feedback, callback) {
  feedback.create_time = Date.now();
  new FeedbackModel(feedback).save(callback);
}
exports.addFeedback = addFeedback;
/*addFeedback({
 "user_id": "576bbe0aa1d183c42c06c08e",
 "phone": "13716962771",
 "content": "不错不错22222!"
 }, callback)*/
```

4) 实现 db.js 的编码

```
/**
 * 数据库操作的总接口模块
 */
var dao_user = require('./dao_user')
var dao_feedback = require('./dao_feedback')

module.exports = {

  getUser: function(phone, callback) {
    dao_user.getUser(phone, function(error, user) {
      if (error) {
        throw error;
      } else {
        callback(user);
      }
    })
  },

  addUser: function(phone, callback) {
```

```
    dao_user.addUser(phone, function(error, user) {
      if (error) {
        throw error;
      } else {
        callback(user);
      }
    })
  },

  addFeedback: function(feedback, callback) {
    dao_feedback.addFeedback(feedback, function(error, feedback) {
      if (error) {
        throw error;
      } else {
        callback(feedback);
      }
    })
  }
};
```

5) 实现分路由的编码

- user.js：用户分路由

```
/**
 * 配置处理用户相关请求的路由模块
 */
var db = require('../db/db')

module.exports = function (router) {
    var users = {}; //保存所有 phone:code 的对象
    /*
    发送验证码短信
     */
    router.get('/sendcode', function (req, res, next) {
        //1. 获取请求参数数据
        var phone = req.query.phone;
        console.log('/sendcode ', phone);
        //2. 处理数据
            //生成验证码(6 位随机数)
        var code = randomCode(6);
            //发送给指定的手机号
        console.log('向' + phone + '发送验证码: '+code);
```

```
        //保存数据(号码:验证码)
    users[phone] = code;
    //3. 返回响应数据
    res.send({"code": 0})
})

/*
 生成指定长度的随机数
 */
function randomCode(length) {
    var chars = ['0','1','2','3','4','5','6','7','8','9'];
    var result = ""; //统一改名: alt + shift + R
    for(var i = 0; i < length ; i ++) {
        var index = Math.ceil(Math.random()*9);
        result += chars[index];
    }
    return result;
}

/*
登陆
 */
router.post('/login', function (req, res, next) {
    //phone=13716962779&code=123123
    var phone = req.body.phone;
    var code = req.body.code;
    console.log('/login', phone, code);

    //检查code是否正确, 如果不正确, 返回{"code" : 1}
    if(users[phone]!=code) {
        res.send({code : 1});
        return;
    }
    //删除保存的code
    delete users[phone];

    //查询, 如果有, 返回, 如果没有添加
    db.getUser(phone, function (user) {
        if(user!=null) {
            res.send({
                "code": 0,
                "data": user
            })
        } else {
```

```
                db.addUser(phone, function (user) {
                    res.send({
                        "code": 0,
                        "data": user
                    })
                })
            }
        })
    })

    /*
    意见反馈
     */
    router.get('/feedback', function (req, res, next) {
        var feedback = JSON.parse(req.query.data);
        db.addFeedback(feedback, function (feedback) {
            res.send({
                code : 0,
                data : feedback
            })
        });
    })
}
```

- address.js: 地址分路由

```
/**
 * 地址相关分路由
 */
var db = require('../db/db');
module.exports = function(router) {

};
```

6) 总路由 index.js 的编码

```
/*
总路由: 注册各个分路由
 */
var express = require('express');
var router = express.Router();

//1. 引入各个分路由
var user = require('./user');
```

```
var address = require('./address');
//2. 注册路由
user(router);
address(router);

module.exports = router;
```

7) 在 app.js 中打开数据库连接

```
// 此文件在启动项目时执行一次，此时可以去获取连接
var connection = require('./db/connection');
connection.connect(); // 连接数据库
// 在关闭项目时，断开连接
app.on('close', function () {
  connection.disconnect();
});
```

8) 修改前台 serverService.js 访问接口

```
// const BASE_URL = 'http://localhost:4000';
const BASE_URL = '';
```

2.8. 收货地址模块功能实现

1) 显示地址列表

- serverService.js

```
/*
得到指定用户的地址列表
 */
function getAddrsByUserId(userId) {
    var defer = $q.defer();
    $http({
        url : '/getAddrsByUserId',
        method : 'GET',
        params : {userId:userId},
    })
        .success(function (result) {
```

```
            defer.resolve(result.data);
        })
        .error(function () {
            alert('获取地址列表失败!');
        })
    return defer.promise;
}
```

- addrManageCtrl.js

```
define(['app', 'storageUtil'], function(app, storageUtil) {
  return app.controller('AddrManageCtrl', ['$scope', '$rootScope', 'serverService',
    function($scope, $rootScope, serverService) {
      $rootScope.appTitle = '地址管理'

      //读取user, 得到userid
      var user = storageUtil.local.getItem(storageUtil.KEYS.USER);
      //发送ajax 请求, 得到对应的地址数组并显示
      serverService.getAddrsByUserId(user._id)
        .then(function(addrs) {
          $scope.addrs = addrs;
        })
    }])
})
```

- addrManage.html

```
<link rel="stylesheet" href="css/templates/addrManage.css">
<div id="addr_main">
  <div id="addr_addrList">
   <div class="addr_addr addr_util" ng-repeat="addr in addrs">
    <div class="addr_addr_left fl" ng-click="toOrderConfirm(addr)">
     <div>
      <p class="addr_name">
       <span ng-bind="addr.contactor"></span>
       <span class="addr_tel" ng-bind="addr.phone"></span>
      </p>
      <p class="addr_name">
       <span ng-bind="addr.address+addr.doorplate"></span>
      </p>
     </div>
    </div>
    <div class="addr_addr_right fr">
     <div class="addr_edit" ng-click="toUpdateUI(addr)">
      <i></i>
     </div>
     <div class="addr_del" ng-click="deleteAddr($index)">
      <i></i>
```

```
    </div>
   </div>
  </div>
 </div>
 <div class="addr_addBtn_div">
  <button class="btn addr_addBtn" ng-click="toAddUI()">新增收货地址</button>
 </div>
</div>
```

2) 删除指定地址

- serverservice.js

```
/*
 删除地址
 */
function deleteAddr(id) {
  var defer = $q.defer();
  $http({
    url : '/deleteAddr',
    method : 'GET',
    params : {_id:id},
  })
    .success(function (result) {
      defer.resolve(result);
    })
    .error(function () {
      alert('删除地址失败!');
    })
  return defer.promise;
}
```

- addrManageCtrl.js

```
//删除指定地址
$scope.deleteAddr = function (index) {
  var addr = $scope.addrs[index];
  if(confirm('确定删除'+addr.contactor+'的地址吗?')) {
    serverService.deleteAddr(addr._id)
      .then(function (result) {
        alert('删除成功');
        //更新页面
        $scope.addrs.splice(index, 1);
      })
```

```
  }
}
```

3) 从百度地图查找地址

- chooseCoordinate.html

```
<link rel="stylesheet" href="css/templates/chooseCoordinate.css">
<div class="cc_main" id="mapDiv">
  <div>
    <input type="text" ng-model="searchName">
    <button ng-click="search()">搜索</button>
  </div>

  <div class="cc_main_map">
    <div class="cc_map" id="cc_map">
      <!--<img src="img/map.png" style="width:100%;height:100%;">-->
    </div>
    <div class="cc_icon">
      <img src="img/position_icon2.png"/>
    </div>
  </div>

  <div class="cc_maplist">
    <ul id="cc_searchAddrListData">
      <li class="lp_searchLi" ng-repeat="item in mapAddrs" ng-click="selectAddr(item)">
        <p class="fl">
          <em></em>
        </p>
        <p class="fl">
          <span ng-bind="item.address"></span>
        </p>
      </li>
    </ul>
  </div>
</div>
```

- mapService

```
/**
 * Created by xfzhang on 2016/11/7.
 * 地图相关请求的服务
 */
define(['app'], function(app) {
```

```
  return app.factory('mapService', ['$http', '$q',
    function($http, $q) {

      /**
       * 加载百度地图api
       * @param containterId 显示地图的容器标签的id
       * @param callback 回调函数名
       */
      function loadMapAPI (containterId, callback) {
        var script = document.createElement("script");
        script.type = "text/javascript";
        script.src =
"http://api.map.baidu.com/api?v=2.0&ak=KM2xAiiLZnVowp1KzcNHeqSxziOXYXGL&callback=" +
callback;
        console.log('------', containterId);
        document.getElementById(containterId).appendChild(script);
      }

      /**
       * 得到附近的地址列表
       * @param point
       */
      function getAroundAddrs (point) {
        var defer = $q.defer();
        var url = 'http://api.map.baidu.com/geocoder/v2/?' +
          'ak=yRrFVK2roUIGRlpcsv7pueLjvU7xO9FE&callback=JSON_CALLBACK' +
          '&location=' + point.lat + ',' + point.lng + '&output=json&pois=1';
        $http.jsonp(url)
          .success(function(data) {
            var result = data.result;
            var cityId = result.cityCode;
            var mapAddrs = [];
            result.pois.forEach(function(item) {
              var address = item.name;
              var lng = item.point.x;
              var lat = item.point.y;
              mapAddrs.push({
                address: address,
                lng: lng,
                lat: lat,
                cityId: cityId
              })
            })
            defer.resolve(mapAddrs);
```

```
          })
          .error(function() {
            alert('请求地图地址失败!');
          })
        return defer.promise;
      }

      return {
        loadMapAPI: loadMapAPI,
        getAroundAddrs: getAroundAddrs
      };
    }]);
})
```

- ChooseCoordinateCtrl.js

```
define(['app', 'storageUtil'], function(app, storageUtil) {
  return app.controller('ChooseCoordinateCtrl', ['$scope', '$rootScope','mapService',
    function($scope, $rootScope, mapService) {
      $rootScope.appTitle = '地图选择地址'

      mapService.loadMapAPI('mapDiv', 'init');

      var inputAddress = storageUtil.session.getItem(storageUtil.KEYS.INPUT_ADDR);

      var map;
      //添加全局函数
      window.init = function() {
        map = new BMap.Map("cc_map");            // 创建Map实例

        var lng = "116.404";
        var lat = "39.915";

        if (inputAddress != null && inputAddress.lng) {
          lng = inputAddress.lng;
          lat = inputAddress.lat;
        }

        var point = new BMap.Point(lng, lat); // 创建点坐标
        map.centerAndZoom(point, 15);
        map.setCurrentCity("北京");

        //监听
        //拖拽结束
        map.addEventListener("dragend", showList);
```

```
        //缩放结束
        map.addEventListener("zoomend", showList);

        //初始显示列表
        showList();
      }

      function showList () {
        //alert('showList');
        //得到中心点坐标对象
        var cPoint = map.getCenter();
        //根据 cPoint 得到附近的多个地址的列表
        mapService.getAroundAddrs(cPoint)
          .then(function(mapAddrs) {
            $scope.mapAddrs = mapAddrs;
          })
      }

      //搜索
      $scope.search = function() {
        var name = $scope.searchName && $scope.searchName.trim()
        if (name) {
          mapService.getPointByAddr(name)
            .then(function(point) {
              map.centerAndZoom(point, 15);
              showList();
            })
        }
      }

      //选择地址
      $scope.selectAddr = function(addr) {
        //保存 addr
        storageUtil.session.setItem(storageUtil.KEYS.MAP_ADDR, addr);
        //跳转
        window.location = '#/add_new_addr';
      }
    }])
})
```

4) 添加地址

- serverService.js

```
/*
 添加地址
 */
function addAddr (addr) {
  var defer = $q.defer();
  $http({
    url : '/insertAddr',
    method : 'GET',
    params : {address:addr},
  })
    .success(function (result) {
      defer.resolve(result.data);
    })
    .error(function () {
      alert('添加地址失败!');
    })
  return defer.promise;
}
```

- AddNewAddrCtrl.js

```
define(['app', 'storageUtil'], function(app, storageUtil) {
  return app.controller('AddNewAddrCtrl', ['$scope', '$rootScope', 'serverService',
    function($scope, $rootScope, serverService) {
      $rootScope.appTitle = '添加地址'

      //准备 address 对象
      $scope.address = {};

      //读取 local 中的 user
      var user = storageUtil.local.getItem(storageUtil.KEYS.USER);
      $scope.address.userId = user._id;

      //读取 session 中的 input_addr
      var inputAddr = storageUtil.session.getItem(storageUtil.KEYS.INPUT_ADDR);
      if (inputAddr != null) {
        $scope.address = inputAddr;
        storageUtil.session.removeItem(storageUtil.KEYS.INPUT_ADDR);
      }

      //读取 session 中的 map_addr
      var mapAddr = storageUtil.session.getItem(storageUtil.KEYS.MAP_ADDR);
      if (mapAddr != null) {
        $scope.address.address = mapAddr.address;
        $scope.address.cityId = mapAddr.cityId;
```

```
        $scope.address.lng = mapAddr.lng;
        $scope.address.lat = mapAddr.lat;
        storageUtil.session.removeItem(storageUtil.KEYS.MAP_ADDR);
      }

      //设置性别
      $scope.setSex = function(sex) {
        $scope.address.sex = sex;
      }

      // 选择地图地址
      $scope.toAddrMap = function() {
        // 保存当前address
        storageUtil.session.setItem(storageUtil.KEYS.INPUT_ADDR, $scope.address);
        // 跳转
        window.location = '#/choose_oordinate';
      }

      // 提交
      $scope.submit = function() {
        serverService.addAddr($scope.address)
          .then(function(addr) {
            alert('保存地址成功!');
            window.location = '#/addr_manage'
          })
      }
    }])
})
```

- addNewAddr.html

```
<link rel="stylesheet" href="css/templates/addNewAddr.css">
<div id="ana_main">

  <p class="ana_linkman">联系人信息</p>
  <div class="ana_fillin">
    <ul>
      <li>
        <p class="fl">姓 名：</p>
        <p class="fr">
          <input type="text" maxlength="20" name="contactor"
            placeholder="请填写收货人的姓名" ng-model="address.contactor">
        </p>
      </li>
      <li>
```

```
        <p class="fl">性 别：</p>
        <p class="fr ana_sex">
      <span class="ana_man" ng-click="setSex(1)">
       <i data-sex="1" ng-class="{chooseBg:address.sex==1}"></i>男士
      </span>
          <span class="ana_girl" ng-click="setSex(2)">
       <i data-sex="2" ng-class="{chooseBg:address.sex==2}"></i>女士
      </span>
        </p>
      </li>
      <li>
        <p class="fl">手 机：</p>
        <p class="fr">
          <input type="text" name="mobilephone" placeholder="请填写收货人的手机号"
                 ng-model="address.phone">
        </p>
      </li>
    </ul>
  </div>

  <p class="ana_addr">地址信息</p>
  <div class="ana_addrDetail">
    <ul>
      <li>
        <div class="fl">学校\大厦\写字楼:</div>
        <div class="fr">
          <i></i>
          <input type="text" name="address" class="ana_searchAddr" placeholder="点击选
择地址"
                 ng-click="toAddrMap()" ng-model="address.address">
        </div>
      </li>
      <li>
        <div class="fl">详细地址:</div>
        <div class="fr">
          <input type="text" class="ana_detailAddr" name="doorplate" placeholder="请填
写门牌号"
                 ng-model="address.doorplate">
        </div>
      </li>
    </ul>
  </div>
  <button class="ana_btn btn" ng-click="submit()">保存</button>
</div>
```

5) 修改指定地址

- mapService.js

```
/*
 更新地址
 */
function updateAddr (addr) {
  var defer = $q.defer();
  $http({
    url : '/updateAddr',
    method : 'GET',
    params : {address:addr},
  })
    .success(function (result) {
      defer.resolve(result);
    })
    .error(function () {
      alert('修改地址失败!');
    })
  return defer.promise;
}
```

- AddrManageCtrl.js

```
define(['app', 'storageUtil'], function(app, storageUtil) {
  return app.controller('AddrManageCtrl', ['$scope', '$rootScope', 'serverService',
    function($scope, $rootScope, serverService) {
      $rootScope.appTitle = '地址管理'

      //删除 session 中的 edit_addr
      storageUtil.session.removeItem(storageUtil.KEYS.EDIT_ADDR);

      //读取 user, 得到 userid
      var user = storageUtil.local.getItem(storageUtil.KEYS.USER);
      //发送 ajax 请求, 得到对应的地址数组并显示
      serverService.getAddrsByUserId(user._id)
        .then(function(addrs) {
          console.log(addrs);
          $scope.addrs = addrs;
        })

      //删除指定地址
```

```
      $scope.deleteAddr = function (index) {
        var addr = $scope.addrs[index];
        if(confirm('确定删除'+addr.contactor+'的地址吗?')) {
          serverService.deleteAddr(addr._id)
            .then(function (result) {
              alert('删除成功');
              //更新页面
              $scope.addrs.splice(index, 1);
            })
        }
      }

      $scope.toAddUI = function () {
        window.location = '#/add_new_addr';
      }

      //更新指定地址
      $scope.toUpdateUI = function (addr) {
        //保存addr到session
        storageUtil.session.setItem(storageUtil.KEYS.EDIT_ADDR, addr)
        window.location = '#/add_new_addr';
      }

      //进入订单确认界面
      $scope.toOrderConfirm = function (addr) {
        storageUtil.session.setItem(storageUtil.KEYS.ORDER_ADDR, addr);
        window.location = '#/order_confirm';
      }
    }])
})
```

- AddNewAddrCtrl.js

```
define(['app', 'storageUtil'], function (app, storageUtil) {
    return app.controller('AddNewAddrCtrl', ['$scope', '$rootScope', 'serverService',
        function ($scope, $rootScope, serverService) {

            //读取session中edit_addr, 依此判断是更新还保存
            var editAddr = storageUtil.session.getItem(storageUtil.KEYS.EDIT_ADDR);
            if(editAddr!=null) { //更新
                $rootScope.appTitle = '更新地址'
                $scope.address = editAddr;
            } else {//增加
                $rootScope.appTitle = '添加地址'
```

```
            //准备 address 对象
            $scope.address = {};
            //读取 local 中的 user
            var user = storageUtil.local.getItem(storageUtil.KEYS.USER);
            $scope.address.userId = user._id;
        }

        //读取 session 中的 input_addr
        var inputAddr = storageUtil.session.getItem(storageUtil.KEYS.INPUT_ADDR);
        if(inputAddr!=null) {
            $scope.address = inputAddr;
            storageUtil.session.removeItem(storageUtil.KEYS.INPUT_ADDR);
        }

        //读取 session 中的 map_addr
        var mapAddr = storageUtil.session.getItem(storageUtil.KEYS.MAP_ADDR);
        if(mapAddr!=null) {
            $scope.address.address = mapAddr.address;
            $scope.address.cityId = mapAddr.cityId;
            $scope.address.lng = mapAddr.lng;
            $scope.address.lat = mapAddr.lat;
            storageUtil.session.removeItem(storageUtil.KEYS.MAP_ADDR);
        }

        //设置性别
        $scope.setSex = function (sex) {
            $scope.address.sex = sex;
        }

        //选择地图地址
        $scope.toAddrMap = function () {
            /*
            $scope.address.address = '龙隆昌科技楼'
            $scope.address.lat = '39.99392711698915';
            $scope.address.lng = '116.32432928208593';
            $scope.address.cityId = 113;
            */
            //保存当前 address
            storageUtil.session.setItem(storageUtil.KEYS.INPUT_ADDR,
$scope.address);



            window.location = '#/choose_oordinate';
        }
```

```
            //提交
            $scope.submit = function () {
                if(editAddr!=null) {
                    serverService.updateAddr($scope.address)
                        .then(function (result) {
                            console.log(result);
                            alert('修改地址成功!');
                            window.location = '#/addr_manage'
                        })
                } else {
                    serverService.addAddr($scope.address)
                        .then(function (addr) {
                            console.log(addr);
                            alert('保存地址成功!');
                            window.location = '#/addr_manage'
                        })
                }

            }
    }])
})
```

2.9. 首页功能实现

1) 功能点:

- 动态获取广告图片, 轮播图展示
- 和菜品列表相关数据, 列表显示
- 购物车操作
- 定位显示当前地址

2) home.html

```
<link rel="stylesheet" href="css/templates/home.css">
<link rel="stylesheet" href="css/swiper.min.css">
<div id="homeDiv">
 <!-- 头部 #start -->
 <header>
  <div class="addr">
   <span class="icon icon-addr"></span>
```

```
   <a href="#/location_addr" ng-bind="data.address.name"></a>
  </div>
  <div class="user">
   <span class="icon icon-user"></span>
   <a href="#/personal">个人中心</a>
  </div>
</header>
<!-- 头部 #end -->

<!-- 轮播 #start -->
<div class="banner">
 <div class="swiper-container" id="bannerSwiper">
  <div class="swiper-wrapper" id="slides">
   <div class="swiper-slide" ng-repeat="banner in banners">
    <a href="#">
     <img src="img/default.jpg" ng-src="{{banner.img_src}}"/>
    </a>
   </div>
  </div>
 </div>

 <!-- Add Pagination -->
 <div class="swiper-pagination"></div>
</div>
<!-- 轮播 #end -->

<!-- 菜品列表 #start -->
<div class="wrap meals">
 <div id="info">
  <ul>
   <li ng-repeat="meal in data.meals">
    <div class="meal-item">
     <a class="img meal-img" href="javascript:">
      <img src="img/default.jpg" ng-src="{{meal.picture}}"/>
     </a>
     <div class="meal-name" ng-bind="meal.mealName"></div>
     <div class="sales" ng-bind="'月售'+meal.sales+'份'"></div>
     <div class="buy">
      <div class="price">
       &yen;
       <span class="ml2 f150 fb" ng-bind="meal.price"></span>
       <span>
        <s ng-bind="'¥'+meal.originalPrice"></s>
       </span>
```

```
        </div>
        <div class="option-cart">
         <div class="btn-cart in fr" ng-class="{incart : meal.count>0}">
          <a class="jian" ng-click="updateMealCount(false, meal, $event)"></a>
          <span class="count" ng-bind="meal.count"></span>
          <a class="jia" ng-click="updateMealCount(true, meal, $event)"></a>
         </div>
        </div>
       </div>
      </div>
     </li>
     <div class="clearfix"></div>
    </ul>
   </div>
  </div>
  <!-- 菜品列表 #end -->

  <!-- 购物车 #start -->
  <footer>
   <div class="blurrycontent"></div>
   <div id="cart" style="position:relative">
    <!-- 购物车图标-->
    <div class="icon-cart-box" ng-click="changeState()">
     <div class="cart-num" id="total_count" ng-bind="cart.totalCount"></div>
     <div class="icon-cart">
      <a class="btn-cart-pop">
      </a>
     </div>
    </div>
    <div class="table w100 cartItems" id="cartItems" ng-show="isOpen">
     <div class="swiper-container" id="cartSwiper">
      <div class="swiper-wrapper">
       <div class="swiper-slide">
        <div class="table w100">
         <div class="row" ng-repeat="meal in cart.meals">
          <div class="cell" ng-bind="meal.mealName"></div>
          <div class="cell">
           &yen;<span class="m12 f120 fb" ng-bind="meal.price"></span>
          </div>
          <div class="cell tr">
           <div class="option-cart fr">
            <div class="btn-cart in fr incart">
             <a class="jian cart-num-btn" ng-click="updateMealCount(false, meal,
$event)"></a>
```

```
            <span class="count" ng-bind="meal.count"></span>
            <a class="jia cart-num-btn" ng-click="updateMealCount(true,
meal,$event)"></a>
          </div>
         </div>
        </div>
       </div>
      </div>
     </div>
    </div>
   </div>
   <!-- Add Scroll Bar -->
   <div class="swiper-scrollbar"></div>

   <div class="bottom">
    <div class="table w100">
     <div class="cell total tc fb f120">
      共计：&yen;<span id="total_price" ng-bind="cart.totalPrice"></span>元
     </div>
     <div class="cell tr">
      <a href="javascript:" class="btn-toPay" ng-class="{able : cart.totalCount>0}"
ng-click="toOrderConfirm()">选好了</a>
     </div>
    </div>
   </div>
  </div>
 </footer>
 <!-- 购物车 #end -->
</div>
```

3) serverService.js

```
/*
 得到轮播图数据
 */
function getBanners() {
  var defer = $q.defer();
  $http({
    url: '/index/banners',
    method: 'GET'
  })
```

```
    .success(function (result) {
      defer.resolve(result.data);
    })
    .error(function () {
      alert('获取轮播图失败!');
    })
  return defer.promise;
}

/*
 得到首页菜品列表,商家,地址相关数据
 */
function getData() {
  var defer = $q.defer();
  $http({
    url: '/index/data',
    method: 'GET'
  })
    .success(function (result) {
      defer.resolve(result.data);
    })
    .error(function () {
      alert('获取菜品信息失败!');
    })
  return defer.promise;
}
```

4) mapService.js

```
/**
 * 得到当前的地址
 * @returns {Promise}
 */
function getCurrentAddr() {
    var defer = $q.defer();
    var geolocation = new BMap.Geolocation();
    geolocation.getCurrentPosition(function (r) {
        if (this.getStatus() === BMAP_STATUS_SUCCESS) {
            //alert('您的位置：'+r.point.lng+','+r.point.lat);
            var point = r.point;

            var geoc = new BMap.Geocoder();
```

```
            geoc.getLocation(point, function (rs) {
                var addComp = rs.addressComponents;
                alert(addComp.province + ", " + addComp.city + ", " + addComp.district
+ ", " + addComp.street + ", " + addComp.streetNumber);
                var name = addComp.city + addComp.district + addComp.street +
addComp.streetNumber;
                defer.resolve({
                    name: name,
                    lng: point.lng,
                    lat: point.lat
                });
            });
        } else {
            alert('定位失败')
        }
    }, {enableHighAccuracy: true})

    return defer.promise;
}
```

5) HomeCtrl.js

```
define(['app', 'Swiper', 'storageUtil'], function (app, Swiper, storageUtil) {
  return app.controller('HomeCtrl', ['$scope', '$rootScope', 'mapService',
'serverService', '$timeout',
    function ($scope, $rootScope, mapService, serverService, $timeout) {
      $rootScope.appTitle = '首页'

      //获取首页相关数据，并显示
      serverService.getBanners()
        .then(function (banners) {
          $scope.banners = banners;

          setTimeout(function () {
            new Swiper('#bannerSwiper', { //配置对象
              pagination: '.swiper-pagination', //指定提示圆点显示的容器选择器
              paginationClickable: true,  //提示圆点是否可点击
              autoplay: 3000, //自动播放的间隔时间
              autoplayDisableOnInteraction: false, //手动操作后重新自动播放
              loop: true, //循环翻页
              effect: 'cube', //翻页效果:立体
```

```
            cube: {
              shadow: false //下部没有阴影
            }
          })
        }, 100); //必须延迟创建Swiper
      })
    serverService.getData()
      .then(function (data) {
        $scope.data = data;

        //在得到data数据之后, 需要比较data中meals与cart中的meals, 并同步
        updateMeals();

        //保存送餐费
        $scope.cart.songcanfei = data.restaurant.songcanfei;
        //保存送商家id
        $scope.cart.rstId = data.restaurant._id;

        //获取保存的定位地址
        var locAddr = storageUtil.session.getItem(storageUtil.KEYS.LOC_ADDR);
        if (locAddr != null) {//直接显示地址
          $scope.data.address = locAddr;
        } else {//定位当前地址
          $scope.data.address.name = '正在定位中...'
          mapService.loadMapAPI('homeDiv', 'initMap');
          window.initMap = function () {
            mapService.getCurrentAddr()
              .then(function (address) {
                console.log(address);
                $scope.data.address = address;
                storageUtil.session.setItem(storageUtil.KEYS.LOC_ADDR, address);
              })
          }
        }
      })

    /*
     更新meals
     */
    function updateMeals() {
      var dMeals = $scope.data.meals;
      var cMeals = $scope.cart.meals;

      for (var i = 0; i < dMeals.length; i++) {
```

```
      var dMeal = dMeals[i];
      for (var j = 0; j < cMeals.length; j++) {
        var cMeal = cMeals[j];
        if (dMeal._id === cMeal._id) {
          //将购物车中meal 的count 给data 中的meal
          dMeal.count = cMeal.count;
          //用data 中的meal 替换cart 中对应的meal
          cMeals[j] = dMeal;
          //cMeal = dMeal;
        }

      }
    }
  }

  //初始化购物车
  initCart();

  function initCart() {
    var cart = storageUtil.session.getItem(storageUtil.KEYS.CART);
    if (cart === null) {
      cart = {
        meals: [],
        "songcanfei": 0,
        "totalPrice": 0,
        "totalCount": 0,
        "rstId": null
      };
    }
    $scope.cart = cart;
  }

  /*
   更新购物车中某个菜品数量
   */
  $scope.updateMealCount = function (isAdd, meal, event) {

    /*
     * 克隆出一个发生事件的<a>, 指定内容1, 指定样式, 添加到<body>
     * 计算出当前的起始坐标(根据<a>): startLeft, startTop, 并设置上
     * 计算目录位置的坐标(id 为total_count 的div): endLeft, endTop
     * 移动动画
     * 持续多长时间: totalTime = 500
     * 小移动的间隔时间: intervalTime = 20
```

```
 * 计算出移动的次数: moveCount = totalTime/intervalTime
 * 计算出每小的移动的距离:
 moveX = (endLeft-startLeft) / moveCount
 moveY = (endTop-startTop) / moveCount
 * 启动循环定时器
 * 更新startLeft/startTop
 * 设置给<a>
 * 判断是否已经到达目标处, 如果到达了, 停止定时器
 */

//克隆出一个发生事件的<a>, 指定内容1, 指定样式, 添加到<body>
var $a = angular.element(event.target);
var $flyA = $a.clone().html('1').addClass('jia-fly');
var $body = angular.element(document.body);
$body.append($flyA);

//计算出当前的起始坐标(根据<a>): startLeft, startTop, 并设置上
var startLeft = $a[0].getBoundingClientRect().left;
var startTop = $a[0].getBoundingClientRect().top;
$flyA.css({
  left: startLeft + 'px',
  top: startTop + 'px'
})

//计算目录位置的坐标(id为total_count的div): endLeft, endTop
var totalCountDiv = document.getElementById('total_count');
var endLeft = totalCountDiv.getBoundingClientRect().left;
var endTop = totalCountDiv.getBoundingClientRect().top;

//移动动画
//持续多长时间: totalTime = 500
var totalTime = 500;
//小移动的间隔时间: intervalTime = 20
var intervalTime = 20;
//计算出移动的次数: moveCount = totalTime/intervalTime
var moveCount = totalTime / intervalTime;
//计算出每小的移动的距离:
var moveX = (endLeft - startLeft) / moveCount;
var moveY = (endTop - startTop) / moveCount;
//启动循环定时器
var intervalId = setInterval(function () {
  //更新startLeft/startTop
  startLeft += moveX;
  startTop += moveY;
```

```
        //判断是否已经到达目标处，如果到达了，停止定时器
        if (startLeft <= endLeft) {
          startLeft = endLeft;
          startTop = endTop;
          clearInterval(intervalId)
        }
        //设置给<a>
        $flyA.css({
          left: startLeft + 'px',
          top: startTop + 'px'
        })
      }, intervalTime);

      $timeout(function () {
        //移除<a>
        $flyA.remove();
        if (isAdd) { //增加
          if (meal.count) { //已经在购物车中了
            meal.count++;
          } else {
            meal.count = 1;
            //添加到购物车
            $scope.cart.meals.push(meal);
          }

          $scope.cart.totalPrice += meal.price;
          $scope.cart.totalCount += 1;
        } else {
          meal.count--;
          if (meal.count === 0) { //从购物车移除
            //找到下标
            var index = $scope.cart.meals.findIndex(function (item, index) {
              return item.count === 0;
            })
            //删除
            $scope.cart.meals.splice(index, 1);
          }

          $scope.cart.totalPrice -= meal.price;
          $scope.cart.totalCount -= 1;
          if ($scope.cart.totalCount === 0) {
            $scope.isOpen = false;
          }
        }
```

```
        //保存cart
        storageUtil.session.setItem(storageUtil.KEYS.CART, $scope.cart);
      }, totalTime + 20);

    }

    $scope.isOpen = false;
    $scope.changeState = function () {
      this.isOpen = !this.isOpen;
    }

    $scope.toOrderConfirm = function () {
      if ($scope.cart.totalCount > 0) {
        window.location = '#/order_confirm'
      }
    }
  }])
})
```

2.10. 设置当前地址

1) locationAddr.html

```
<link rel="stylesheet" href="css/templates/locationAddr.css">

<div id="loc_addr_div">
  <div class="lp_search">
    <div class=" table w100 ">
      <div class="cell" style="width:1.5rem;"><i></i>
      </div>
      <div class="cell">
        <input type="text" id="search_addr" placeholder="请输入所在写字楼/小区/学校等"
               ng-model="searchText" ng-change="search()"/>
      </div>
      <div class="cell tr">
        <a href="javascript:" id="cancel" class="btn"
           ng-show="showSearch" ng-click="cancelSearch()">取消</a>
      </div>
    </div>
  </div>
```

```
  <div class="lp_searchAddrList" ng-show="showSearch">
    <ul id="lp_searchAddrListData">
      <li class="addrItem" ng-repeat="searchAddr in searchAddrArr"
          ng-bind="searchAddr.name" ng-click="toIndex(searchAddr)">
      </li>
    </ul>
  </div>

  <div class="addr-wrap" id="info" ng-hide="showSearch">
    <p class="p-label">当前位置</p>
    <div class="addr cur-addr table w100">
      <div class="cell addrItem" ng-bind="locationAddr.name"
ng-click="toIndex(locationAddr)"></div>
      <div class="lp_position cell tr" ng-click="relocation()">
        <a id="doPos" href="javascript:"><i ng-class="{posing:locating}"></i>重新定位
</a>
      </div>
    </div>
    <p class="p-label">附近地址</p>
    <div class="addr">
      <ul>
        <li class="addrItem" ng-repeat="aroundAddr in aroundAddrArr"
            ng-bind="aroundAddr.address" ng-click="toIndex(aroundAddr)"></li>
        <li style="position:relative;">
          <a style="display:block;width:100%;height:100%;" href="#/choose_oordinate">
            更多地址 <span class="fr nav-left"
style="position:absolute;right:0;top:50%;margin-top:-5px;"></span>
          </a>
        </li>
      </ul>
    </div>
    <p class="p-label">收货地址</p>
    <div class="addr">
      <ul>
        <li class="addrItem" ng-repeat="address in addressArr"
ng-click="toIndex(address)">
          <p class="user" ng-bind="address.contactor +' '+address.phone"></p>
          <p class="user-addr" ng-bind="address.address+address.doorplate"></p>
        </li>
      </ul>
    </div>
  </div>
</div>
```

2) LocationAddrCtrl.js

```
define(['app', 'storageUtil'], function (app, storageUtil) {
  return app.controller('LocationAddrCtrl', ['$scope', '$rootScope', 'serverService',
'mapService',
    function ($scope, $rootScope, serverService, mapService) {

      $rootScope.headTitle = "选择地址";

      //检查用户是否登陆
      var user = storageUtil.local.getItem(storageUtil.KEYS.USER);
      if (user == null) {
        alert('请先登陆!');
        window.location = '#/login';
        return;
      }

      //显示地址列表
      serverService.getAddrsByUserId(user._id)
        .then(function (addressArr) {
          $scope.addressArr = addressArr;
        });

      //显示当前地址
      var locationAddr = storageUtil.session.getItem(storageUtil.KEYS.LOC_ADDR);
      $scope.locationAddr = locationAddr;
      //请求获取当前地址附近地址列表显示
      mapService.getAroundAddrs(locationAddr)
        .then(function (addrs) {
          $scope.aroundAddrArr = addrs;
        });

      //根据地址关键字搜索相关的地址列表
      $scope.showSearch = false;
      $scope.search = function () {
        var searchText = $scope.searchText;
        if (searchText.trim() === '') {
          return;
        }

        $scope.showSearch = true;
        mapService.getPlaceAddrs(searchText)
          .then(function (addrs) {
```

```
            $scope.searchAddrArr = addrs;
          })
      }

      //取消搜索
      $scope.cancelSearch = function () {
        $scope.showSearch = false;
        $scope.searchText = '';
        $scope.searchAddrArr = null;
      }


      $scope.locating = false;
      //重新定位
      $scope.relocation = function () {
        $scope.locating = true;

        mapService.loadMapAPI('loc_addr_div', 'initMap');
        window.initMap = function () {
          mapService.getCurrentAddr()
            .then(function (address) {
              $scope.locationAddr = address;
              $scope.locating = false;
              return mapService.getAroundAddrs(address)   //它会作为then()的返回值
            })
            .then(function (addrs) {
              $scope.aroundAddrArr = addrs;
            });
        }
      };

      /*
       进入首页
       */
      $scope.toIndex = function (address) {
        var addr = {
          name: address.address || address.name,
          lat: address.lat,
          lng: address.lng
        };
        storageUtil.session.setItem(storageUtil.KEYS.LOC_ADDR, addr);
        window.location = '#/home';
      }
    }])
```

```
})
```

2.11. 订单相关功能实现

1) 功能点

- 订单确认界面，下单
- 显示订单详情
- 显示当前用户的订单列表

2) OrderConfirm.html

```
<link rel="stylesheet" href="css/templates/orderConfirm.css">

<div id="oc_main">
  <div class="oc_addr">
    <a href="#/addr_manage">
      <p>
        <span class="oc_userName" ng-bind="orderAddress.contactor"></span>
        <span class="oc_tel" ng-bind="orderAddress.phone"></span>
      </p>
      <p class="oc_userAddr"
ng-bind="orderAddress.address+orderAddress.doorplate"></p>
      <i></i>
    </a>
  </div>
  <div class="oc_sendTime">
    <p class="fl">送餐时间:</p>
    <div class="fl oc_time">
      <select id="oc_time" ng-model="order.arrive_time">
        <option value="">未选择</option>
        <option ng-repeat="time in times"
value="{{time.value}}">{{time.text}}</option>
      </select>
    </div>
    <p class="fl oc_estimate">预计 1 小时后送达</p>
  </div>
  <div class="oc_label">订单明细</div>
  <div class="oc_orderDetail">
    <div class="oc_food">
```

```
      <dl ng-repeat="meal in cart.meals">
        <dt class="fl">
          <img ng-src="{{meal.picture}}"/>
        </dt>
        <dd class="fl">
          <p class="oc_foodName fl" ng-bind="meal.mealName"></p>
          <p class="oc_foodPric fr" ng-bind="meal.price+'元'"></p>
          <p class="oc_foodNum fr" ng-bind="'X'+meal.count"></p>
        </dd>
      </dl>
    </div>

    <div class="oc_remarks">
      <p>备注</p>
      <textarea class="oc_remark" ng-model="order.remark"></textarea>
    </div>
  </div>
  <div class="oc_label">费用小计</div>
  <div class="oc_cost">
    <ul>
      <li>
        <span class="fl">配送费用</span>
        <span class="fr oc_delivery" ng-bind="cart.songcanfei | currency:'￥'"></span>
      </li>
      <li>
        <span class="fl">费用合计</span>
        <span class="fr oc_total" ng-bind="cart.totalPrice | currency:'￥'"></span>
      </li>
    </ul>
  </div>
  <div class="oc_btn_div">
    <button class="oc_btn oc_ybtn btn" href="javascript:" ng-click="submit()">微信支
付</button>
  </div>
</div>
```

3) serverService.js

```
/*
 获取用户的默认地址
 */
function getDefaultAddr(userId) {
```

```
  var defer = $q.defer();
  $http({
    url: '/order/getNewestAddress?userId=' + userId,
    method: 'GET'
  })
    .success(function (result) {
      defer.resolve(result.data); //如果没有匹配的, 传入的 undefined
    })
    .error(function () {
      alert('请求默认地址失败!');
    })
  return defer.promise;
}

/*
 下单
 */
function addOrder(order) {
  var defer = $q.defer();
  $http({
    url: '/order/createOrder',
    method: 'POST',
    data: {order: order}
  })
    .success(function (result) {
      defer.resolve(result.data);
    })
    .error(function () {
      alert('下单失败!');
    })
  return defer.promise;
}
```

4) OrderConfirmCtrl.js

```
define(['app', 'storageUtil'], function (app, storageUtil) {
  return app.controller('OrderConfirmCtrl', ['$scope', '$rootScope', 'serverService',
'$filter',
    function ($scope, $rootScope, serverService, $filter) {
      $rootScope.appTitle = '下 单'

      //初始化 order
```

```
$scope.order = {};

//检查用户是否登陆,如果没有提示登陆
var user = storageUtil.local.getItem(storageUtil.KEYS.USER);
if (user == null) {
  alert('请先登陆!')
  window.location = '#/login'
  return;
}

//从 session 取出保存的用户地址, 如果有, 显示
var orderAddress = storageUtil.session.getItem(storageUtil.KEYS.ORDER_ADDR);
//如果没有发送获取当前用户的默认地址的 ajax, 如果有, 显示
if (orderAddress == null) {
  serverService.getDefaultAddr(user._id)
    .then(function (address) {
      if (!!address) {
        $scope.orderAddress = address;
      } else {
        //没有
        alert('没有一个地址,先添加!');
        window.location = '#/add_new_addr';
      }
    })
} else {
  $scope.orderAddress = orderAddress;
}

//送餐时间列表
initTimes();

//读取购物车数据, 并显示
var cart = storageUtil.session.getItem(storageUtil.KEYS.CART);
$scope.cart = cart;



/*
 使用$filter 的 date 过滤器来格式化日期时间
 */
function initTimes() {
  var times = [];   //{value:'2016-11-9 12:10', text:'11:10'}
  var startTime = new Date().getTime();
  times.push({
    value: $filter('date')(startTime + 60 * 60 * 1000, 'yyyy-MM-dd HH:mm'),
```

```
        text: '立即配送'
      })
      var endTime = new Date($filter('date')(startTime, 'yyyy-MM-dd') + '
20:00').getTime();
      var intervalTime = 15;//单位是min
      while (startTime <= endTime) {
        //更新startTime
        startTime += intervalTime * 60 * 1000;
        //如果超过了结束循环
        if (startTime > endTime) {
          break;
        }
        times.push({
          value: $filter('date')(startTime + 60 * 60 * 1000, 'yyyy-MM-dd HH:mm'),
          text: $filter('date')(startTime, 'HH:mm')
        })
      }
      $scope.times = times;
    }


    /*
     手动拼日期时间
     */
    function initTimes2() {
      var times = [];   //{value:'2016-11-9 12:10', text:'11:10'}

      var date = new Date();
      var year = date.getFullYear();
      var month = date.getMonth() + 1;
      var day = date.getDate();
      var hour = date.getHours();
      var minite = date.getMinutes();
      //添加第一个
      times.push({
        value: year + '-' + month + '-' + day + ' ' + (hour + 1) + ':' + minite,
        text: '立即配送'
      })

      //添加后面的多个
      var startTime = date.getTime();
      var endTime = new Date(year + '-' + month + '-' + day + ' 20:00').getTime();
      var intervalTime = 15;//单位是min
      while (startTime <= endTime) {
```

```
        //更新startTime
        startTime += intervalTime * 60 * 1000;
        //如果超过了结束循环
        if (startTime > endTime) {
          break;
        }

        minite += intervalTime;
        if (minite >= 60) {
          hour++;
          minite -= 60;
        }
        times.push({
          value: year + '-' + month + '-' + day + ' ' + (hour + 1) + ':' + minite,
          text: hour + ':' + minite
        })
      }

      $scope.times = times;

    }

    $scope.submit = function () {

      /*
       {
       "user_id": "576bbe0aa1d183c42c06c08e",

       "contactor": "张晓飞",
       "address": "龙隆昌科技楼",
       "phone": "13716962779",
       "doorplate": "3层301",

       "total_money": 56,
       "peisongfei": 0,

       "remark": "加一份米饭",
       "arrive_time": "2016-6-23 20:14",

       "detail":
"{\"data\":{\"rstId\":1772,\"money\":56,\"meals\":[{\"mealName\":\"藜麦牛油果沙拉
\",\"pictures\":\"%2Fsource%2Fassets%2Fimages%2Fimg%2F3.png\",\"num\":1,\"price\":\
"23\"},{\"mealName\":\"牛油果三文鱼沙拉
\",\"pictures\":\"%2Fsource%2Fassets%2Fimages%2Fimg%2F5.png\",\"num\":1,\"price\":\
```

```
"33\"}]}}"
        }
        */

        $scope.order.user_id = user._id;

        $scope.order.contactor = $scope.orderAddress.contactor;
        $scope.order.address = $scope.orderAddress.address;
        $scope.order.phone = $scope.orderAddress.phone;
        $scope.order.doorplate = $scope.orderAddress.doorplate;

        $scope.order.total_money = $scope.cart.totalPrice;
        $scope.order.peisongfei = $scope.cart.songcanfei;
        var dataObj = {
          rstId: $scope.cart.rstId,
          money: $scope.order.total_money + $scope.order.peisongfei,
          meals: $scope.cart.meals
        }
        $scope.order.detail = JSON.stringify({data: dataObj}); //detail 是个字符串

        serverService.addOrder($scope.order)
          .then(function (order) {
            console.log(order);
            alert('下单成功');

            storageUtil.session.removeItem(storageUtil.KEYS.CART);
            storageUtil.session.removeItem(storageUtil.KEYS.ORDER_ADDR);

            window.location = '/order/detail?id=' + order._id;
          })
      }
    }])
})
```

orderDetail.ejs

```
<!DOCTYPE html>
<html>

<head>
 <meta charset="UTF-8">
 <title>订单详情</title>
 <meta content="telephone=no" name="format-detection">
 <meta content="width=device-width, initial-scale=1.0, maximum-scale=1.0,
user-scalable=no" name="viewport">
```

```
<meta content="yes" name="apple-mobile-web-app-capable">
<meta content="black" name="apple-mobile-web-app-status-bar-style">
<meta content="on" http-equiv="cleartype">

<link rel="stylesheet" type="text/css" href="/css/reset.css" />
<link rel="stylesheet" type="text/css" href="/css/layout.css" />
<link rel="stylesheet" type="text/css" href="/css/common.css" />

<style>
 #od_main{
  width: 100%;
  height: 100%;
 }
 .od_schedule{
  width: 97%;
  margin: 0 auto;
  overflow: hidden;
  background-color: #fff;
 }
 .od_schedule_top{
  width: 100%;
  overflow: hidden;
  padding:22px 25%;
  border-bottom: 1px solid #f0f0f0;
 }
 .od_schedule_top dl{

 }
 .od_schedule_top dt{
  margin: 0 auto;
  width: 2.5rem;
  height: 2.81rem;
  margin-right: 13px;
 }
 .od_business{
  font-size: 1.25rem;
  line-height: 1.93rem;
 }
 .od_time{
  font-size: 0.75rem;
  color: #969696;
 }
 .od_time span{
  color: #000;
```

```
}
.od_schedule_bottom{
 width: 100%;
 padding: 25px 0;
 position: relative;
}
.od_schedule_line{
 width: 65%;
 margin: 0 auto;
 overflow: hidden;
}
.od_bgr{
 background: #eb3c3c;
}
.od_bgg{
 background: #c7c7c7;
}
.od_schedule_line div{
 width: 50%;
 height: 0.13rem;
}
.od_schedule_font{

 text-align: center;
}
.od_schedule_font ul{
 margin: 0 auto;
 margin-bottom: 7px;
}
.od_schedule_font ul li{
 display: table-cell;
 width: 20rem;
 text-align: center;
}
.od_schedule_line span{
 position: absolute;
 bottom: 25%;
}
.od_schedule_line span:nth-child(1){
 left: 15%;
}
.od_schedule_line span:nth-child(2){
 left: 49%;
}
```

```
.od_schedule_line span:nth-child(3){
 left: 81%;
}
.od_schedule_line span,i{
 display: inline-block;
}
.od_schedule_line i{
 width: 0.56rem;
 height: 0.56rem;
 display: inline-block;
 border-radius: 50%;
}
.od_schedule_y{
 border:2px solid #eb3c3c;
 width: 0.94rem;
 height: 0.94rem;
 line-height: 0.63rem;
 text-align: center;
 border-radius: 50%;
}
.od_detail{
 width: 97%;
 margin: 0 auto;
 margin-top: 15px;
 background-color: #fff;
 position: relative;
 margin-bottom: 3.2rem;

}
.od_detail_top{
 padding-left: 15px;
 overflow: hidden;
 width: 100%;
 padding-top: 30px;
}
.od_detail_top .od_foodPic{
 width: 20%;
 height: 3.7rem;
}
.od_foodDetail p:nth-child(1){
 width: 68%;
}
.od_detail_top .od_foodPic img{
 border:1px solid #d9d9d9;
```

```
}
.od_foodDetail{
 width: 74%;
 margin-left: 10px;
 margin-top: 2%;
 margin-bottom: 3.5rem;
}
.od_detail_mid{
 font-size: 0.88rem;
 color: #333333;
 border-bottom: 1px solid #f0f0f0;
}
.od_detail_mid ul{
 width: 95%;
 overflow: hidden;
 margin: 0 auto;
 margin-top: 10px;
}
.od_detail_mid ul:first-child{
 border-top: 1px dashed #d9d9d9;
}
.od_detail_mid ul li{
 overflow: hidden;
 height: 1.88rem;
 line-height: 1.88rem;
 border-bottom: 1px dashed #d9d9d9;
 padding-left: 10px;
}
.od_noborder{
 border-bottom: none !important;
}
.od_detail_bottom{
 margin: 9px;
 font-size: 0.88rem;
 padding-left: 10px;
}
.od_detail_bottom p{
 line-height: 24px;
 color: #333333;
}
.od_label{
 position: absolute;
 width: 4.5rem;
 height: 1.88rem;
```

```
 background: #eb3c3c;
 line-height: 30px;
 text-align: center;
 color: #fff;
 top:-5px;
}
.od_meal_detail{
 overflow: hidden;
}

.od_footer{
 width: 100%;
 height: 3.125rem;
 line-height: 3.125rem;
 background: #fff;
 position: fixed;
 bottom: 0;
 padding: 0 0.625rem 0 1.25rem ;
}
.od_footer div{
 display: inline-block;
 margin-top:0.43rem;
}
.od_tel{
 width: 2.25rem;
 height: 2.25rem;
}
.od_backOrder{
 width: 2.25rem;
 height: 2.25rem;
 line-height: 2.25rem;
 text-align: center;
 color:#000;
 border-radius: 50%;
 border:1px solid #646464;
 margin-left: 1rem;
}

.od_sureOrder button{
 width: 6.88rem;
 height: 2.34rem;
 background: #1e8c5a;
 border: none;
 display: block;
```

```
   text-align: center;
   color: #fff;
  }
  .load_div{
   position: absolute;
   top: 40%;
   left: 50%;
  }
  .od_dsn{
   display: none;
  }
  @media only screen and  (min-width: 320px) and  (max-width: 480px) {
  html{
     font-size: 17px;
  }
 }
 </style>
</head>

<body>

 <div id="od_main">
  <div class="od_schedule">
   <div class="od_schedule_top">
    <dl>
     <dt class="fl">
      <img src="/img/od_schedule.png" />
     </dt>
     <dd class="fl">
      <p class="od_business">
       <%=stateText%>
      </p>
      <p class="od_time">预计送达：<span><%=arrive_time%></span></p>
     </dd>
    </dl>
   </div>
    <div class="od_schedule_bottom">


    <%
     if(state==5) {
    %>
     <!--state=5-->
     <div class="od_schedule_font">
```

```
      <ul>
       <li>已付款</li>
       <li>商家已接单</li>
       <li>确认送达</li>
      </ul>
     </div>
     <div class="od_schedule_line">
      <span class="od_schedule_y"><i class="od_schedule_n od_bgr"></i></span>
      <span class="od_schedule_y"><i class="od_schedule_n od_bgr"></i></span>
      <span class=""><i class="od_schedule_n od_bgg"></i></span>
      <div class="od_line_left fl  od_bgg"></div>
      <div class="od_line_right fr od_bgg"></div>
     </div>
    <%
     } else if(state==1) {
    %>
     <!--state=1-->
     <div class="od_schedule_font">
      <ul>
       <li>待接单</li>
       <li>商家已接单</li>
       <li>确认送达</li>
      </ul>
     </div>
     <div class="od_schedule_line">
      <span class="od_schedule_y"><i class="od_schedule_n od_bgr"></i></span>
      <span class=""><i class="od_schedule_n od_bgg"></i></span>
      <span class=""><i class="od_schedule_n od_bgg"></i></span>
      <div class="od_line_left fl  od_bgg"></div>
      <div class="od_line_right fr od_bgg"></div>
     </div>
    <%
     } else if(state==3) {
    %>
     <!--state=3-->
     <div class="od_schedule_font">
      <ul>
       <li>已付款</li>
       <li>商家已接单</li>
       <li>确认送达</li>
      </ul>
     </div>
     <div class="od_schedule_line">
      <span class="od_schedule_y"><i class="od_schedule_n od_bgr"></i></span>
```

```
      <span class="od_schedule_y"><i class="od_schedule_n od_bgr"></i></span>
      <span class="od_schedule_y"><i class="od_schedule_n od_bgr"></i></span>
      <div class="od_line_left fl  od_bgg"></div>
      <div class="od_line_right fr od_bgg"></div>
     </div>
    <%
     } else if(state==6) {
    %>
     <!--state=6-->
     <div class="od_schedule_font">
      <ul>
       <li>申请取消</li>
       <li>店铺处理</li>
       <li>订单取消</li>
      </ul>
     </div>
     <div class="od_schedule_line">
      <span class="od_schedule_y"><i class="od_schedule_n od_bgr"></i></span>
      <span class="od_schedule_y"><i class="od_schedule_n od_bgr"></i></span>
      <span class="od_schedule_y"><i class="od_schedule_n od_bgr"></i></span>
      <div class="od_line_left fl  od_bgg"></div>
      <div class="od_line_right fr od_bgg"></div>
     </div>
    <%
     }
    %>
   </div>
</div>
<div class="od_detail">
 <div class="od_detail_top">
  <div class="od_label">订单详情</div>

  <%
   var meals = detail.data.meals;
   for (var i = 0; i < meals.length; i++) {
    var meal = meals[i];
  %>
    <div class="od_meal_detail">
     <div class="od_foodPic fl">
      <img src="<%='/'+meal.picture%>" />
     </div>
     <div class="od_foodDetail fl">
      <p class="fl"><%=meal.mealName%></p>
      <p class="fl">X<%=meal.count%></p>
```

```
        <p class="fr"><%=meal.price%>元</p>
       </div>
      </div>
    <%
     }
    %>



  </div>
  <div class="od_detail_mid">
   <ul>
    <li>
     <p class="fl">配送费用</p>
     <p class="fr"><%=peisongfei%>元</p>
    </li>
    <li class="od_noborder">
     <p class="fl">费用合计</p>
     <p class="fr"><%=detail.data.money%>元</p>
    </li>
   </ul>
  </div>
  <div class="od_detail_bottom">
   <p>配送地址：<%=address+doorplate%></p>
   <p>收货电话：<%=phone%></p>
  </div>
 </div>

 <div class="od_footer">
  <div class="od_tel fl">
   <a href="tel:13111111111"><img src="/img/om_tel.png" /></a>
  </div>

  <div class="od_backOrder fl">
   退单
  </div>

  <div class="od_sureOrder fr" >
   <button class="od_sureOrder_btn" onclick="window.location='/'">确认送达</button>
  </div>
 </div>
</div>
</body>
</html>
```

3.项目构建

3.1. 下载 grunt 及其插件包

```
"devDependencies": {
  "grunt": "^1.0.1",
  "grunt-contrib-clean": "^1.0.0",
  "grunt-contrib-concat": "^1.0.1",
  "grunt-contrib-copy": "^1.0.0",
  "grunt-contrib-cssmin": "^1.0.2",
  "grunt-contrib-htmlmin": "^2.0.0",
  "grunt-contrib-imagemin": "^1.0.1",
  "grunt-contrib-jshint": "^1.0.0",
  "grunt-contrib-requirejs": "^1.0.0",
  "grunt-contrib-uglify": "^2.0.0",
  "grunt-contrib-watch": "^1.0.0"
}
```

3.2. 配置 Gruntfile.js

```
/**
 * 此配置文件本质就是一个 node 函数模块
 */
module.exports = function (grunt) {
  //1. 初始化插件配置
  grunt.initConfig({

    //清除文件的任务
    clean: ["build/css", "build/js", "build/index.html"],

    //打包 requirejs 管理的所有模块的任务
    requirejs: {
      compile: {
        options: {
          name: 'main',    //主配置文件名字(不能带后缀)
          mainConfigFile: 'client/js/main.js',    //require 的主文件路径
          out: 'build/js/main.js'            //压缩后的文件
        }
      }
    },
```

```
    //批量、无损压缩图片的任务
    imagemin: {
      dist: {
        options: {
          optimizationLevel: 3 //定义 PNG 图片优化水平
        },
        files: [{
          expand: true, // 设置为true，表示要支持cwd等更多配置
          cwd: 'client/img/', //基本路径(只针对src有效，对dest无效)
          src: ['*.{png,jpg,jpeg,gif}'], // 优化目录下所有png/jpg/jpeg/gif图片
          dest: 'build/img/' // 优化后的图片保存位置
        }]
      }
    },

    //js 语法检查的任务
    jshint: {
      options: {
        jshintrc: '.jshintrc' //语法检查的配置文件
      },
      files: ['Gruntfile.js', 'client/js/**/*.js', '!client/js/libs/*.js'] //检查哪些
文件
    },

    //合并压缩css的任务
    cssmin: {
      target: {
        files: [
          {
            expand: true,
            cwd: 'client/css/templates',
            src: ['*.css'],
            dest: 'build/css/templates'
          },
          {
            //输出文件: 被所有被处理文件的数组
            'build/css/output.min.css': [
              'client/css/*.css',
              '!client/css/swiper.min.css'
            ]
          }
        ]
      }
```

```
  },

  //压缩 html 页面的任务
  htmlmin: {
    dist: {
      options: {
        removeComments: true,  //删除注释
        collapseWhitespace: true //合并多行空行
      },
      files: [
        {
          expand: true, // 设置为 true,表示要支持 cwd 等更多配置项
          cwd: 'client/js/templates/', //基础目录
          src: ['*.html'], //将要被压缩的所有文件的数组
          dest: 'build/js/templates' //输出目标文件夹
        },
        {
          expand: true,
          cwd: 'client/',
          src: ['index.html'],
          dest: 'build/'
        }
      ]
    }
  },

  //拷贝文件的任务
  copy: {
    main: {
      //将不需要处理的文件直接从源目录拷贝到目标目录中
      files: [
        {
          src: ['client/js/libs/require.min.js'],
          dest: 'build/js/require.min.js'
        },
        {
          src: ['client/css/swiper.min.css'],
          dest: 'build/css/swiper.min.css'
        }
      ],
    },
  },

  //监视
```

```
    watch: {
      scripts: {
        files: ['client/js/templates/*.html', 'client/js/**/*.js',
          '!client/js/libs/*.js', 'client/css/*.css'],
        tasks: ['requirejs', 'cssmin', 'htmlmin'],
        options: {spawn: false} //增量更新, 默认为true 表示全量更新
      }
    },
  });

  //2. 加载插件任务
  grunt.loadNpmTasks('grunt-contrib-jshint');
  grunt.loadNpmTasks('grunt-contrib-clean');
  grunt.loadNpmTasks('grunt-contrib-imagemin');
  grunt.loadNpmTasks('grunt-contrib-cssmin');
  grunt.loadNpmTasks('grunt-contrib-htmlmin');
  grunt.loadNpmTasks('grunt-contrib-requirejs');
  grunt.loadNpmTasks('grunt-contrib-copy');
  grunt.loadNpmTasks('grunt-contrib-watch');

  //3. 注册构建任务
  grunt.registerTask('dev',
    ['jshint', 'clean', 'requirejs', 'imagemin',
      'cssmin', 'htmlmin', 'copy', 'watch']);
};
```

3.3. 配置.jshintrc

```
{
  "curly": true,
  "eqeqeq": true,
  "eqnull": true,
  "expr" : true,
  "immed": true,
  "newcap": true,
  "noempty": true,
  "noarg": true,
  "regexp": true,
  "browser": true,
  "devel": true,
  "node": true,
```

```
  //使用未定义的变量
  "undef": true,

  //语句后面没有分号
  "asi": true,
  "boss": true,
  //预定义的全局变量(不检查)
  "predef": [ "define", "BMap", "angular", "BMAP_STATUS_SUCCESS"]
}
```

3.4. 修改 main.js

```
// baseUrl: 'js/',
```

3.5. 修改 app.js

```
// app.use(express.static(path.join(__dirname, 'public')));
app.use(express.static(path.join(__dirname, 'build')));
```

3.6. 修改 index.html

```
<link rel="stylesheet" type="text/css" href="css/output.min.css" />
<script type="text/javascript" src="js/require.min.js" data-main="js/main"></script>
```

3.7. 修改 orderDetail.ejs 与 orderList.ejs

```
<link rel="stylesheet" type="text/css" href="css/output.min.css" />
```

4. 总结

基于 Angular 及其相关插件的应用开发
基于 Node+Mongodb 的后台开发
基于百度地图的地图定位应用开发
基于 RequireJS 的模块化开发
基于 Grunt 的项目自动化构建