第7章 智慧环保项目

07

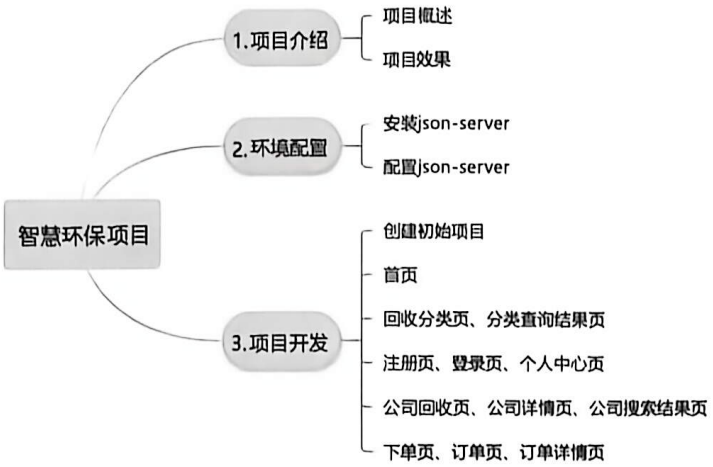
本章导读

本章以智慧环保项目为例介绍 uni-app 开发的流程，综合应用 uni-app 的组件、API、扩展组件 uView 等，智慧环保项目包括首页、回收分类页、分类查询结果页等各种页面。

学习目标

知识目标	1. 掌握 uView 2.0 的使用 2. 掌握 API 的使用 3. 掌握 flex 布局
能力目标	1. 能够查看 uView 的官方文档 2. 能够理解 scss 代码 3. 能够实现常规的移动页面开发 4. 具有使用 uView 开发实用项目的能力
素质目标	1. 具有积极探索新知、自主学习与钻研的精神 2. 具有节能控制、绿色发展的理念 3. 具有精益求精、自主解决问题的素质 4. 具有团队协作、善于沟通的素质

知识思维导图



7.1 项目介绍

7.1.1 项目概述


智慧环保项目

如今，城市化速度越来越快，社会发展形态也在发生变化，因此，我国提出了创造智慧城市、提高城市管理水平以及提供多元化城市服务的发展战略。在此背景下，人们利用各种先进信息技术，有效整合城市各项配套系统和功能模块，进一步促进城市朝工业化、信息化、城镇化方向发展。党的二十大报告指出“加快构建废弃物循环利用体系”“打造宜居、韧性、智慧城市”。城市环境保护工作有序开展，本项目实现一个简洁版的智慧环保系统，主要包括的页面如下。

（1）首页：展示环境宣传信息和快捷功能。

（2）订单页：登录后，用户可以查看订单信息。普通用户可以查询待接单、已接单、已完成、已取消的订单信息，并可以修改订单状态。企业用户可以查看相应的订单信息、处理客户提交的订单信息。

（3）公司回收页：用户可以查看附近的回收公司，还可以查看回收公司的详情。

（4）个人中心页：展示登录用户的个人信息。

（5）回收分类页：让用户了解哪些垃圾属于可回收的，以便于选择回收类型。

（6）分类查询结果页：显示回收分类页中点击“搜索”按钮后的结果。

（7）注册页和登录页：当处于需要登录才能查看的页面时，如未登录，会跳转到登录页。注册页用于注册新用户。

（8）下单页：选择需要回收的垃圾的类别，填写预约上门信息，让工作人员上门服务。

（9）公司搜索结果页：显示在公司回收页中点击“搜索”按钮后的结果。

（10）公司详情页：显示单个回收公司的详细信息。

（11）订单详情页：显示单个订单的详细信息。

7.1.2 项目效果

（1）首页

首页包括搜索框、轮播图、快捷功能、回收操作流程、【预约上门回收】按钮、爱心活动等，如图 7-1 所示。点击快捷功能图标，将进入对应页面。点击【预约上门回收】按钮，将进入公司回收页。

（2）订单页

在已登录的情况下，用户可在订单页查看所有待接单信息，如图 7-2 所示。点击某一订单，将进入相应的订单详情页。点击顶部的【待接单】【已接单】【已完成】【已取消】，将展示对应的订单列表。如果未登录，打开订单页会跳转到登录页。

图 7-1 首页

图 7-2 订单页

（3）公司回收页

该页面显示所有的回收公司信息，如图 7-3 所示。在此页面上方的搜索框输入相应关键字进入公司搜索结果页，点击某一公司信息，将进入相应的公司详情页。

（4）个人中心页

在未登录的情况下，个人中心页显示未登录。如已登录将显示用户的相关信息，如图 7-4 所示。点击【签到】，则可打开日历进行签到。点击快捷功能，将进入对应的功能页面。

图 7-3　公司回收页

图 7-4　个人中心页

（5）回收分类页

点击首页中的【回收分类】，进入回收分类页，如图 7-5 所示。在该页面左侧点击大类，该页面右侧显示相应的子类信息。

（6）分类查询结果页

在首页的搜索框或本页的搜索框中输入查询关键字进行查询，可进入分类查询结果页，如图 7-6 所示。

图 7-5　回收分类页

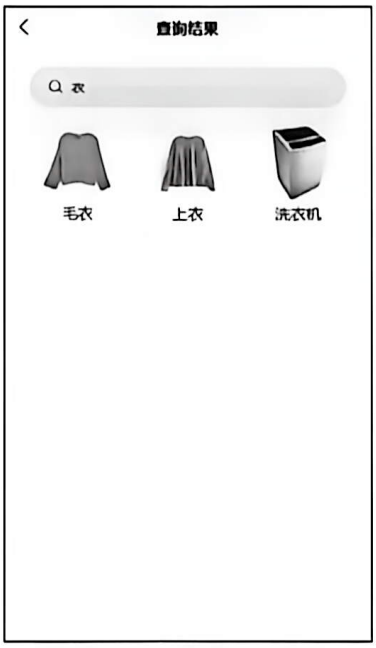


图 7-6　分类查询结果页

（7）注册页和登录页

要查看订单、预约上门回收等都必须先登录才可以进行。如果没有登录，则会跳转到登录页。如果没有注册，则会跳转到注册页。注册页如图 7-7 所示，登录页如图 7-8 所示。

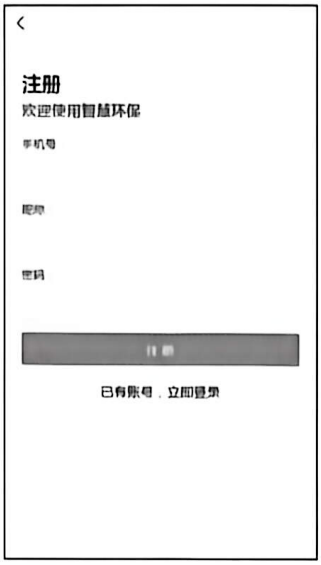


图 7-7　注册页

图 7-8　登录页

（8）下单页

在公司回收页点击【立即下单】按钮，或在公司详情页中点击【预约上门回收】按钮，会跳转到下单页，如图 7-9 所示。在输入相关信息后，点击【立即预约】按钮，如果预约成功，则跳转到订单页，可查看提交的订单信息。

（9）公司搜索结果页

在公司回收页的搜索框中输入关键字进行搜索，可进入公司搜索结果页，如图 7-10 所示。

图 7-9　下单页

图 7-10　公司搜索结果页

（10）公司详情页

在公司回收页的公司列表中点击公司列表项，则会打开公司详情页，如图 7-11 所示。

（11）订单详情页

在订单页中点击订单列表项，则会打开订单详情页，如图 7-12 所示。

图 7-11　公司详情页

图 7-12　订单详情页

7.2　环境配置

7.2.1　安装 json-server

开发项目的时候，如果后端接口还没有创建出来，前端工程师也不必等后端接口创建出来才进行下一步开发。可以使用 mock.js 来模拟接口数据。这种用于模拟后端接口数据的工具有很多，例如 fastmock、lazy-mock、api-fox、json-srever 等。本项目采用 json-server 工具。json-server 是一个存储 JSON 数据的服务器，用来模拟后端接口数据。

在命令提示符窗口中，运行下列命令（要先安装 Node.js，参考 2.2 节）。

```
npm install -g json-server
```

可以通过查看版本号来测试 json-server 是否安装成功。

```
json-server --version
```

在任意地方新建一个文件夹（提供的源代码中为 smartEpserver 文件夹），进入该文件夹，在地址栏中输入“cmd”并按“Enter”键，则在命令提示符窗口中进入该文件夹。执行下列命令启动一个 Web 服务器，如图 7-13 所示。

```
json-server --watch db.json
```

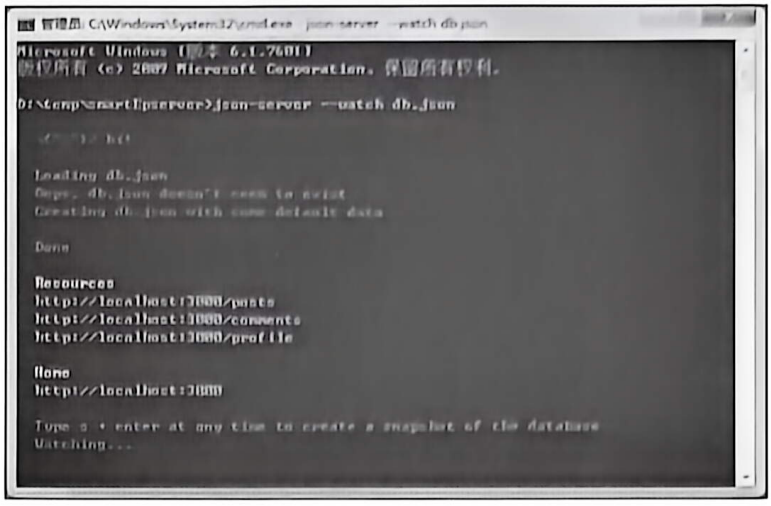


图 7-13　启动一个 Web 服务器

执行以上命令后，可以看到新建的文件夹中多了一个 db.json 文件，里面有一些默认数据。在浏览器中输入 http://localhost:3000，可以打开对应页面，如图 7-14 所示。这样服务器就建成了，里面有 3 个子页面地址，这就是模拟的接口地址。访问这些地址，可以显示 db.json 文件中对应的内容。

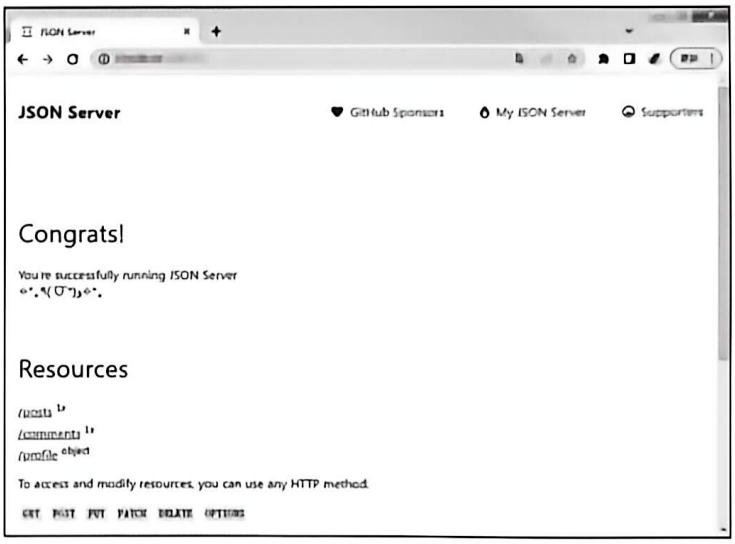


图 7-14　页面效果

7.2.2　配置 json-server

如果不想用 3000 端口，可以在启动服务器时指定需要的端口。以下命令用于将服务器端口改为 3004。

```
json-server --watch db.json --port 3004
```

在 smartEpserver 的根目录下新建一个 package.json 文件，将启动命令写到该文件中，代码如下。

```
{
    "scripts":{       "mock":"json-server db.json --port 3004"    }
}
```

以后启动服务器可用 npm run mock 命令。如果要停止服务器，只需关闭命令提示符窗口。

在 smartEpserver 的根目录下新建一个 public 文件夹，将图片资源放入 public 文件夹。修改 db.json 文件，具体代码如下所示。这里由于篇幅限制，每一种类型的记录仅保留了一条，其他省略。

```
{
  "types":[
 {    "id":1,
      "parentId":0,
      "icon":"",
      "name":"废纸",
      "unitPrice":0
    },
    ……省略

    {   "id":101,
      "parentId":1,
      "icon":"/image/type/101.jpg",
      "name":"报纸",
      "unitPrice":10
    },
    ……省略
  ],
  "users":[
{   "id":1,
      "phoneNum":"1567×××××××",
      "nickName":"xiaoyu",
      "password":"123456",
      "avatar":"/image/avatar/3.jpeg",
      "typeid":1
    },
   ……省略
  ],
  "hotDots":[
{     "id":1,
      "image":"/image/swt1.jpg",
      "title":""
    },
    ……省略
  ],
  "actions":[
{   "id":1,
      "content":"红牌洗衣液可以洗羊毛衫、羽绒服、棉织品等。",
      "coverUrl":"/image/xiyiye.jpg",
      "title":"200 积分限时兑换 500mL 浓缩洗衣液，快来抢购！"
    },
    ……省略
```

```
  ),
  "companys":[
    {   "id":1,
      "address":"湖北省武汉市光谷民族大道下线村 100 米处",
      "contact":"1841XXXXXXX",
      "coverUrl":"/image/company1.jpg",
      "introduction":"再生资源回收以物资不断循环……",
      "name":"信昌废旧物资回收公司",
      "score":1
    },
    ……省略
  ],
  "orders":[
    {    "id":1,
      "companyName":"时代回收公司",
      "createby":"小明",
      "createTime":"2022-03-15 10:35:58",
      "pickupTime":"14:00—16:00",
      "pickupAddress":"武汉市江夏区蜜糖镇",
      "goodNumber":3,
      "goodUnitPrice":2,
      "goodTypeName":"牛仔裤",
      "coverUrl":"/image/1.jpg",
      "contact":"1567×××××××",
      "remark":"",
      "pkgs":12,
      "state":1,
      "companyId":3,
      "userId":2
    },
    ……省略
  ]
}
```


通过访问 http://localhost:3004/users 可以得到 users 里面的所有记录。若访问成功，返回的 res 的结构如图 7-15 所示，其中返回的数据 res.data 为数组。有关 json-server 更多的知识，请参考本书的电子资源。


```
▼{data: Array(1), statusCode: 200, header: {…}, errMsg: "request:ok"}
```


图 7-15　res 的结构

本项目的接口如表 7-1 所示。

表 7-1　接口

序号	名称	接口	请求方式
1	用户列表	http://localhost:3004/users	GET
2	订单列表	http://localhost:3004/orders	GET
3	分类列表	http://localhost:3004/types	GET

续表

序号	名称	接口	请求方式
4	轮播宣传图片列表	http://localhost:3004/hotDots	GET
5	活动列表	http://localhost:3004/actions	GET
6	公司列表	http://localhost:3004/companys	GET
7	注册用户	http://localhost:3004/users	POST
8	下单	http://localhost:3004/orders	POST
9	取消订单	http://localhost:3004/orders/订单编号	PATCH

7.3 项目开发

7.3.1 创建初始项目

（1）新建项目

用默认模板新建一个 Vue2 的 uni-app 项目 smartEpApp。新建 3 个 tabBar 页面（已自动生成一个 index.vue 页面），分别为订单页 order.vue、公司回收页 company.vue、个人中心页 me.vue。修改 pages.json 文件，增加 tabBar 节点，实现底部导航栏。pages.json 的代码如下。

```
{
    "pages":[{
        "path":"pages/index/index",
        "style":{    "navigationBarTitleText":"诚 信 回 收",
            "navigationBarBackgroundColor":"#00c297",
            "navigationBarTextStyle":"white"
        }
    }, {
        "path":"pages/company/company",
        "style":{    "app-plus":{
                "titleNView":false
            }
        }
    }, {
        "path":"pages/order/order",
        "style":{    "navigationBarTitleText":"我的订单",
            "enablePullDownRefresh":false
        }

    }, {
        "path":"pages/me/me",
        "style":{    "navigationBarTitleText":"个人中心"
        }

    }],
    "globalStyle":{
"navigationBarTextStyle":"black",
        "navigationBarTitleText":"uni-app",
```

```
        "navigationBarBackgroundColor":"#F8F8F8",
        "backgroundColor":"#F8F8F8",
        "app-plus":{    "background":"#efeff4"    }
    },
    "tabBar":{
        "color":"#909399",
        "selectedColor":"#303133",
        "backgroundColor":"#FFFFFF",
        "borderStyle":"black",
        "list":[{
                "pagePath":"pages/index/index",
                "iconPath":"static/tabbar/home.png",
                "selectedIconPath":"static/tabbar/home-selected.png",
                "text":"首页"
            },{
                "pagePath":"pages/order/order",
                "iconPath":"static/tabbar/order.png",
                "selectedIconPath":"static/tabbar/order-selected.png",
                "text":"订单"
            },{
                "pagePath":"pages/company/company",
                "iconPath":"static/tabbar/company.png",
                "selectedIconPath":"static/tabbar/company_selected.png",
                "text":"公司回收"
            },    {
                "pagePath":"pages/me/me",
                "iconPath":"static/tabbar/user.png",
                "selectedIconPath":"static/tabbar/user-selected.png",
                "text":"我的"
            } ]
    }
}
```

其中，company.vue 页面使用代码“"app-plus": {"titleNView": false}”隐藏了导航栏。可以通过设置“"navigationStyle": "custom"”，使用自定义导航栏，以关闭默认的原生导航栏。

```
{
      "path":"pages/company/company",
      "style":{
          "navigationStyle":"custom"
          }
   }
```

下面将其他页面配置一并放到 pages 中。

```
"pages":[
.       ...省略前面的 tabBar 页面
, {     "path":"childPages/type/recyclingMenu",
        "style":{    "navigationBarTitleText":"",
              "enablePullDownRefresh":false
        }
    },{ "path":"childPages/type/searchRes",
        "style":{    "navigationBarTitleText":"",
              "enablePullDownRefresh":false
```

```
        )
    ),{ "path":"pages/login/login",
        "style":{    "navigationBarTitleText":"",
              "enablePullDownRefresh":false
        )
    ),{ "path":"pages/login/register",
        "style":{    "navigationBarTitleText":"",
              "enablePullDownRefresh":false
        )
    ),{ "path":"childPages/company/detail",
        "style":{    "navigationBarTitleText":"诚 信 回 收",
              "navigationBarBackgroundColor":"#00c297",
              "navigationBarTextStyle":"white"
        )
    ),{ "path":"childPages/company/searchRes",
        "style":{    "navigationBarTitleText":"搜索结果",
              "navigationBarBackgroundColor":"#00c297",
              "navigationBarTextStyle":"white"
        )
    ),{ "path":"childPages/company/fillIn",
        "style":{    "navigationBarTitleText":"",
              "enablePullDownRefresh":false
        )
    ),,{   "path" :"childPages/order/waiting",
          "style" :{    "navigationBarTitleText":"",
              "enablePullDownRefresh":false
          )
      )]
```

（2）安装插件

在插件市场搜索前端组件 uView，使用 HBuilderX 导入插件的方式安装 uView 2.0。安装完成后，项目根目录下多了一个 uni_modules 文件夹，uView 的组件都在/uni_modules/uview-ui 文件下面。

uView 依赖 scss，必须安装 scss 插件，uView 才能正确运行。在插件市场找到“scss/sass 编译”插件，进行安装。如果已安装，则略过此步。

（3）配置 uView

引入 uView 主 JS 库。在项目根目录中的 main.js 中，添加以下代码，引入并使用 uView 的主 JS 库。需要注意的是，这两行代码要放在“import Vue from 'vue'”的后面。

```
import uView from '@/uni_modules/uview-ui'
Vue.use(uView)
```

引入主题文件。在项目根目录下的 uni.scss 文件中引用 uView 的全局 scss 主题文件 theme.scss。

```
@import '@/uni_modules/uview-ui/theme.scss';
```

引入 uView 样式文件。在 App.vue 中的<style></style>部分，引入 uView 基础样式文件 index.scss。注意，如果还引用了其他样式文件，引用该样式文件的代码需要写在第 1 行，同时在<style></style>标签中添加“ lang="scss"”。具体代码如下。

```
<style lang="scss">
    @import "@/uni_modules/uview-ui/index.scss";/* 注意要写在第 1 行 */
</style>
```

配置 casycom。在项目根目录的 pages.json 中添加 casycom 字段。如果是通过 HBuiderX 导入插件的方式安装的 uView，可以忽略此配置。

```
"easycom":{
    "^u-(.*)":"@/uni_modules/uview-ui/components/u-$1/u-$1.vue"
},
```

（4）配置全局 baseUrl

在 smartEpApp 根目录下新建 common 文件夹，在 common 文件夹下新建文件 api.js，代码如下。

```
const baseUrl = 'http://localhost:3004';
export default {    baseUrl }
```

将服务器 URL 在 main.js 中进行全局挂载。在 main.js 中创建 Vue 实例之前，添加以下代码。

```
import baseUrl from '@/common/api.js'
Vue.prototype.$baseUrl = baseUrl
```

这样在页面中可以用 this.$baseUrl.baseUrl 来获取 http://localhost:3004。

（5）页面的文件路径如表 7-2 所示。

表 7-2　页面的文件路径

序号	页面	文件路径
1	首页	pages/index/index.vue
2	订单页	pages/order/order.vue
3	公司回收页	pages/company/company.vue
4	个人中心页	pages/me/me.vue
5	回收分类页	childPages/type/recyclingMenu.vue
6	分类查询结果页	childPages/type/searchRes.vue
7	登录页	pages/login/login.vue
8	注册页	pages/login/register.vue
9	公司详情页	childPages/company/detail.vue
10	公司搜索结果页	childPages/company/searchRes.vue
11	下单页	childPages/company/fillIn.vue
12	订单详情页	childPages/order/waiting.vue

7.3.2　首页

首页共有 6 个部分：搜索框、轮播图、快捷功能、回收操作流程、【预约上门回收】按钮、爱心活动。轮播图、爱心活动的数据来源于后端服务器，效果如图 7-1 所示，其请求地址如下，具体数据格式参见 7.2.2 小节的 db.json 文件。

- 轮播图的请求地址：http://localhost:3004/hotDots。
- 爱心活动信息的请求地址：http://localhost:3004/actions。

首页完整代码如下。

```
<template>
    <view>
        <view class="search">
            <u-search placeholder="请输入搜索内容" height="50" shape="square"
v-model="keyword" @search="search"    :showAction="false"></u-search>
        </view>
        <view class="swiper">
      <u-swiper :list="swiperlist" keyName="image" :autoplay="true" circular>
</u-swiper>
        </view>
        <view class="char">
            <view class="char_item" @click="goto(1)">
   <image mode="widthFix" src="../../static/icons/Recycling_classification.png">
</image>
                <view>回收分类</view>
            </view>
            <view class="char_item">
                <image mode="widthFix" src="../../static/icons/Old_things.png">
</image>
                <view>旧物去向</view>
            </view>
            <view class="char_item">
         <image mode="widthFix" src="../../static/icons/Recycling_machine.png">
</image>
                <view>附近回收</view>
            </view>
            <view class="char_item">
      <image mode="widthFix" src="../../static/icons/Shopping_Mall.png"></image>
                <view>积分商城</view>
            </view>
        </view>
        <view class="btnOrder">
            <u-button color="rgb(0, 194, 151)" size="large" shape="circle"
@click="goto(5)">预约上门回收</u-button>
        </view>
        <view class="char char_m">
            <view class="char_item">
                <u-icon name="clock" size="20" label="在线预约"></u-icon>
                <view>第 1 步</view>
            </view>
            <view class="char_item">
                <u-icon name="bell" size="20" label="免费上门"></u-icon>
                <view>第 2 步</view>
            </view>
            <view class="char_item">
                <u-icon name="attach" size="20" label="订单完成"></u-icon>
                <view>第 3 步</view>
```

```
            </view>
            <view class="char_item">
                <u-icon name="star" size="20" label="用户福利"></u-icon>
                <view>第 4 步</view>
            </view>
        </view>
        <view class="activeWrap">
            <view class="title">爱心活动</view>
            <view class="actlist">
                <view class="act_item" v-for="(item,index) in activelist"
:key="index">
    <u-image :src="$baseUrl.baseUrl+item.coverUrl" mode="aspectFill" :border-
radius="20" width="100%"   height="300rpx"></u-image>
                    <text>{{item.title}}</text>
                </view>
            </view>
        </view>
    </view>
</template>
<script>
    export default {
        data() {
            return {
                swiperlist:[],
                keyword:'',
                activelist:[]
            }
        },
        onLoad() {
            this.getHotDots();
            this.getActlist();
        },
        methods:{
            getHotDots() {
                uni.request({
                    url:this.$baseUrl.baseUrl + '/hotDots',
                    methods:'get',
                    success:(res) => {
                        console.log(res.data);
                        this.swiperlist = res.data;
                        for (let i = 0; i < this.swiperlist.length; i++) {
                    this.swiperlist[i].image = this.$baseUrl.baseUrl + this.
swiperlist[i].image;
                        }
                    }
                })
            },
            search(e) {
                console.log(e);
                uni.navigateTo({
                    url:'/childPages/type/searchRes?txtSearch=' + e
                });
                this.keyword = "";
```

```
            ),
            goto(i) {
                switch (i) {
                    case 1:
                        uni.navigateTo({   url:'/childPages/type/recyclingMenu'   });
                        break;
                    case 5:
                        uni.switchTab({      url:'/pages/company/company'    })
                        break;
                    default:
                        break;
                } },
            getActlist() {
                uni.request({
                    url:this.$baseUrl.baseUrl + '/actions',
                    methods:"get",
                    success:(res) => {
                        console.log(res);
                        this.activelist = res.data;
                        console.log(this.activelist);
                    }
                })
            }
        }
    }
</script>
<style lang="scss" scoped>
    .search {
        width:100%;
        box-sizing:border-box;
        padding:20px 20px 15px 20px;
        background-color:rgb(0, 194, 151);
    }
    .swiper {
        padding:10px;
        background-color:#F4F4F5;
    }
    .char {
        margin-top:20px;
        margin-left:10rpx;
        display:flex;
        justify-content:space-around;
    }
    .char_item {
        flex:1;
        text-align:center;
        font-size:14px;
        color:#888888;
    }
    .char_item image {
        width:60%;
    }
    .btnOrder  {padding:20px;    }
    .char_m {    margin-top:0;    }
```

```
        .activeWrap {    padding:0 20px;    }
        .activeWrap .title {
            font-size:16px;
            padding-left:10px;
            border-left:3px solid rgb(0, 194, 151);
            margin-top:10px;
            margin-bottom:10px;
        }
        .activeWrap .actlist {    width:100%;    }
        .act_item {      margin-bottom:10px;  }
    </style>
```

【代码解析】

（1）搜索框

搜索框使用了 uView 中的 search 组件，该组件双向绑定了 data 属性 keyword，按“Enter”键或者手机软键盘右下角的“搜索”键时，触发@search 事件。search 组件可以开启右边按钮，可将该按钮设置为“搜索”或“取消”等内容。:showAction="false"表示右侧的“搜索”按钮不显示。

```
    <u-search placeholder="请输入搜索内容" height="50" shape="square" v-model=
"keyword" @search="search"    :showAction="false"></u-search>
```

JS 代码中，在 data 处添加属性 keyword: ''，在 methods 处添加 search 方法。

（2）轮播图

轮播图使用了 uView 中的 u-swiper 组件。其中 list 属性值为轮播数据，keyName 为 list 数组中对象的目标属性名，autoplay 用于设置是否自动播放，circular 用于设置是否循环播放。

```
    <u-swiper :list="swiperlist" keyName="image" :autoplay="true" circular>
</u-swiper>
```

JS 代码中，在 data 处添加属性 swiperlist: [ ]，在 methods 处添加 getHotDots，onLoad 生命周期函数调用 getHotDots 方法。

（3）快捷功能

这里有 4 个快捷，若点击【回收分类】，执行 goto 方法，跳转到回收分类页。因为后面的代码还没有实现，读者在调试时，应先将对应代码注释掉。快捷功能在布局上使用了 flex 布局。

（4）【预约上门回收】按钮

该按钮使用了 u-Button 组件，color 属性用于设置背景颜色；type 属性用于设置不同颜色的按钮样式，如 default、primary、success、info、waring、error；size 用于设置按钮大小，这里将其设置为 large（大按钮）；shape 用于设置按钮是 circle（圆角）还是 square（直角）。点击此按钮执行 goto 方法。

（5）回收操作流程

与快捷功能一样，回收操作流程使用 flex 布局。这里用了 u-icon 图标，u-icon 是基于字体的图标集，包含大多数场景的图标。其中，name 表示图标名称，读者可以在 uni-app 官网查看图标集。size 用于设置图标大小，默认单位为 px，默认值为 16px。label 用于设置图标周围的文字，可以用 labelPos 设置文字的位置，其取值为 left、top、right、bottom。

```
    <u-icon name="clock" size="20" label="在线预约"></u-icon>
```

（6）爱心活动

爱心活动的数据来源于后端服务器，具体的数据格式可查看 7.2.2 小节的 db.json 文件。爱心活动的效果如图 7-16 所示。这里显示图片使用了 uView 的 u-image 组件，其中图片的 URL 是一个相对地址，其前面需要添加“http://localhost:3004”，可用全局属性$baseUrl.baseUrl 获得。

```
    <u-image :src="$baseUrl.baseUrl+item.coverUrl" mode="aspectFill" :border-
radius="20" width="100%"    height="300rpx"></u-image>
```

图 7-16　爱心活动的效果

7.3.3　回收分类页、分类查询结果页

这两个页面都用于对分类进行操作，是首页的子页面。这里将页面放在 childPages/type 目录下，为项目的分包处理提供支持，参见 2.1.3 小节。

回收分类页的效果如图 7-5 所示，分类查询结果页的效果如图 7-6 所示。分类的数据来源于后端服务器，数据格式请参考 7.2.2 小节中 db.json 文件的 types 属性。其 URL 为 http://localhost:3004/types。

1. 回收分类页

对于回收分类页的手风琴式的导航效果，uView 中提供了一个模板。读者可以在 HBuilderX 插件安装页面搜索“uView”，通过导入示例来获得相关代码，然后在此基础上根据数据格式进行修改，以显示回收分类。除此之外还在页面顶部添加了搜索框。它与首页中的搜索框类似，只是为了页面的美观，在其外面添加 view 组件，设置 CSS 样式。该页面的完整代码如下。

```
<template>
    <view class="u-wrap">
        <view class="u-search-box">
            <view class="u-search-inner">
```

```
                <u-search placeholder="请输入搜索内容" height="30" shape="round"
v-model="keyword" @search="search"    :showAction="false"></u-search>
            </view>
        </view>
        <view class="u-menu-wrap">
            <scroll-view  scroll-y  scroll-with-animation  class="u-tab-view
menu-scroll-view" :scroll-top="scrollTop">
                <view v-for="(item,index) in parentlist" :key="index" class=
"u-tab-item" :class="[current==index ? 'u-tab-item-active' :'']"
    :data-current="index" @tap.stop="swichMenu(index)">
                    <text class="u-line-1">{{item.name}}</text>
                </view>
            </scroll-view>
            <block v-for="(item,index) in parentlist" :key="index">
                <scroll-view scroll-y class="right-box" v-if="current==index">
                    <view class="page-view">
                        <view class="class-item">
                            <view class="item-title">
                                <text>{{item.name}}</text>
                            </view>
                            <view class="item-container">
        <view class="thumb-box" v-for="(item1, index1) in childlist[current]"
:key="index1">
    <image class="item-menu-image" :src="$baseUrl.baseUrl+item1.icon" mode="">
</image>
            <view class="item-menu-name">{{item1.name}}</view>
                            </view>
                        </view>
                    </view>
                </scroll-view>
            </block>
        </view>
    </view>
</template>
<script>
    export default {
        data() {
            return {
                scrollTop:0, //左侧菜单的滚动条位置
                current:0, // 预设当前菜单项的值
                menuHeight:0, // 左侧菜单的高度
                menuItemHeight:0, // 左侧菜单项的高度
                parentlist:[], //左侧菜单的数据
                childlist:[],//右侧的数据与左侧菜单的数据通过数组索引进行对应
                keyword:''
            }
        },
        created() {
            this.getTypelist();
        },
        methods:{
            search(e) {
```

```
                console.log(e);
                uni.navigateTo({
                    url:'/childPages/type/searchRes?txtSearch=' + e
                });
                this.keyword = "";
            },
            getTypelist() {
                console.log(11);
                uni.request({
                    url:this.$baseUrl.baseUrl + "/types?parentId=0",
                    method:"GET",
                    success:(res)=>{
                    if (res.statusCode == 200) {
                        this.parentlist = res.data;
                        for(let i=0; i<this.parentlist.length; i++){
                            this.getChildlist(this.parentlist[i].id);
                        }
                    }
                    }
                })
            },
            getChildlist(id) {
                uni.request({
                    url:"http://localhost:3004/types?parentId="+id,
                    method:"GET",
                    success:(res)=>{
                        console.log(res.data);
                    if (res.statusCode == 200) {
                        this.childlist.push(res.data);
                    }
            }
                })
            },
            getImg() {
                return Math.floor(Math.random() * 35);
            },
            // 点击左边的菜单项切换回收分类
            async swichMenu(index) {
                if(index == this.current) return ;
                this.current = index;
                // 如果菜单或菜单项的高度为 0，意味着尚未初始化
                if(this.menuHeight == 0 || this.menuItemHeight == 0) {
                    await this.getElRect('menu-scroll-view', 'menuHeight');
                    await this.getElRect('u-tab-item', 'menuItemHeight');
                }
                // 将菜单中的菜单项垂直居中
                this.scrollTop = index * this.menuItemHeight + this.menuItemHeight
2 - this.menuHeight / 2;
            },
            // 获取一个目标元素的高度
            getElRect(elClass, dataVal) {
                new Promise((resolve, reject) => {
                    const query = uni.createSelectorQuery().in(this);
```

```
                query.select('.' + elClass).fields({size:true},res => {
                    // 如果节点尚未生成，res 值为 null，循环执行
                    if(!res) {
                        setTimeout(() => {
                            this.getElRect(elClass);
                        }, 10);
                        return ;
                    }
                    this[dataVal] = res.height;
                }).exec();
            })
        }
    }
</script>
<style lang="scss" scoped>
    .u-wrap {
        height:calc(100vh);
        /* #ifdef H5 */
        height:calc(100vh - var(--window-top));
        /* #endif */
        display:flex;
        flex-direction:column;
    }
    .u-search-box { adding:18rpx 30rpx;    }
    .u-search-inner {
        background-color:rgb(234, 234, 234);
        border-radius:100rpx;
        display:flex;
        align-items:center;
        padding:10rpx 10rpx;
    }
    .u-menu-wrap {
        flex:1;
        display:flex;
        overflow:hidden;
    }
    .u-tab-view {
        width:200rpx;
        height:100%;
    }
    .u-tab-item {
        height:110rpx;
        background:#f6f6f6;
        box-sizing:border-box;
        display:flex;
        align-items:center;
        justify-content:center;
        font-size:26rpx;
        color:#444;
        font-weight:400;
        line-height:1;
    }
    .u-tab-item-active {
```

```
        position:relative;
        color:#000;
        font-size:30rpx;
        font-weight:600;
        background:#fff;
    }
    .u-tab-item-active::before {
        content:"";
        position:absolute;
        border-left:4px solid rgb(0, 194, 151);
        height:32rpx;
        left:0;
        top:39rpx;
    }
    .u-tab-view {   height:100%;   }
    .right-box {   background-color:rgb(250, 250, 250);   }
    .page-view {   padding:16rpx;    }
    .class-item {
        margin-bottom:30rpx;
        background-color:#fff;
        padding:16rpx;
        border-radius:8rpx;
    }
    .item-title {
        font-size:26rpx;
        color:$uni-text-color;
        font-weight:bold;
    }
    .item-menu-name {
        font-weight:normal;
        font-size:24rpx;
        color:$uni-text-color;
    }
    .item-container {
        display:flex;
        flex-wrap:wrap;
    }
    .thumb-box {
        width:33.333333%;
        display:flex;
        align-items:center;
        justify-content:center;
        flex-direction:column;
        margin-top:20rpx;
    }
    .item-menu-image {
        width:120rpx;
        height:120rpx;
    }
</style>
```

【代码解析】

回收分类页的层级关系通过 parentId 来体现。其中 parentId 为 0 的项为左侧菜单项，

parentId 不为 0，则表示该项为 id 为 parentId 类的子项。这里首先查询 parentId 为 0 的项，将它们存储在 parentlist 数组中，同时查询该项的子项，将其所有的子项存储在 childlist 数组中，childlist 为一个二维数组。子项和其父项通过数组索引进行关联。

在 JS 代码中，getChildlist(id)方法用于将对应子项存入 childlist 数组。getTypelist 方法用于查找 parentId 为 0 的项，然后调用 getChildlist 查找其子项。

2. 分类查询结果页

分类查询结果页显示包含关键字的所有子项和包含关键字的大类的子项，如图 7-6 所示。当搜索不到相关子项时，显示“暂无搜索数据!”的提示，如图 7-17 所示。

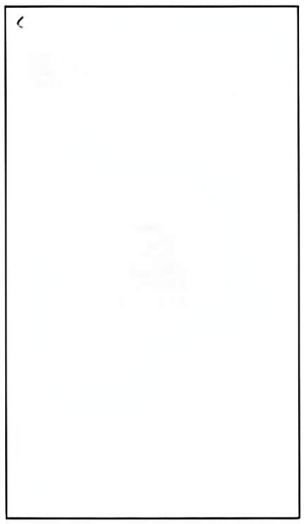


图 7-17　搜索不到相关子项时的演示效果

分类查询结果页的完整代码如下。

```
<template>
    <view class="page-view">
        <view class="u-search-box">
            <view class="u-search-inner">
                <u-search placeholder="请输入搜索内容" height="30" shape="round"
v-model="keyword" @search="search"       :showAction="false"></u-search>
            </view>
        </view>
        <view class="class-item">
            <view class="item-container" v-if="childlist.length>0">
                <view class="thumb-box" v-for="(item,index) in childlist"  :key=
"index" >
                    <image mode="widthFix" :src="$baseUrl.baseUrl+item.icon">
</image>
                    <view class="item-menu-name">{{item.name}}</view>
                </view>
            </view>
            <view class="emptyBox" v-else>
                <u-empty  text="暂无搜索数据!" mode="data"></u-empty>
            </view>
        </view>
    </view>
</template>
<script>
    export default {
        data() {
            return {
                childlist:[],
                keyword:''
            }
        },
        onLoad(options) {
            console.log(options.txtSearch)
```

```
            this.getTypelist(options.txtSearch);
            this.keyword = options.txtSearch;
        },
        methods:{
    search(e) {
        console.log(e);
        uni.navigateTo({
            url:'/childPages/type/searchRes?txtSearch=' + e
        });
        this.keyword = "";
    },
            getTypelist(str) {
                uni.request({
                        url:this.$baseUrl.baseUrl + "/types?name_like="+str,
                        method:"GET",
                        success:(res)=>{
                    console.log(res.data);
                    if(res.statusCode == 200 && res.data.length>0) {
                        let i = 0;
                        for(i = 0; i< res.data.length;i++){
                        if(res.data[i].parentId == 0) {
                            this.getChildlist(res.data[i].id);
                        } else {
                            this.childlist.push(res.data[i]) ;
                        }
                        }
                    }
                    }
                })
            },
            getChildlist(id) {
                uni.request({
                    url:"http://localhost:3004/types?parentId="+id,
                    method:"GET",
                    success:(res)=>{
                    if(res.statusCode == 200) {
                        for(let i  = 0 ;i< res.data.length; i++){
                            this.childlist.push(res.data[i]);
                        }
                        }
                    }
                });
            }
        }
    }
</script>
<style scoped>
    .page-view{    padding:20rpx;    }
    .u-search-box {    padding:18rpx 30rpx;    }
    .u-search-inner {
        background-color:rgb(234, 234, 234);
        border-radius:100rpx;
```

```
        display:flex;
        align-items:center;
        padding:10rpx 10rpx;
    }
    .class-item{
        margin-bottom:30rpx;
        background-color:#fff;
        padding:16rpx;
        border-radius:8rpx;
    }
    .item-container{
        display:flex;
        flex-wrap:wrap;
    }
    .thumb-box{
        width:33.33%;
        text-align:center;
    }
    .thumb-box image{    width:70%;  }
    .emptyBox{
        position:fixed;
        top:50%;
        left:50%;
        transform:translate3d(-50%,-50%,0);
    }
</style>
```

【代码解析】

在<template></template>中若 childlist.length > 0，则显示搜索到的所有子项，否则显示“暂无搜索数据!”。

7.3.4 注册页、登录页、个人中心页

1. 注册页

注册页用来注册新用户，用户类型分普通用户和企业用户，页面效果如图 7-7 所示。本实例主要实现的是普通用户的功能。注册成功会弹出一个提示框，然后自动登录系统，并打开个人中心页。

注册功能用 post 方法请求 http://localhost:3004/users。

```
<template>
    <view class="registerWrap">
        <!-- 标题 -->
        <view class="title">
            <h3>注册</h3>
            <h4>欢迎使用智慧环保</h4>
        </view>
        <!-- 表单 -->
        <view class="formWrap">
            <view class="formItem">
```

```
                <view class="label">手机号:</view>
                <view class="inputBox">
                    <input type="number" v-model="phoneNum" />
                </view>
            </view>
            <view class="formItem">
                <view class="label">昵称:</view>
                <view class="inputBox">
                    <input type="text" v-model="nickName" />
                </view>
            </view>
            <view class="formItem">
                <view class="label">密码:</view>
                <view class="inputBox">
                    <input password="true" v-model="password" />
                </view>
            </view>
        </view>
        <view>
<u-button type="primary" @click="submitReg" color="rgb(0,194,151)">注 册</u-button>
        </view>
        <navigator url="/pages/me/login">
            <view class="registTips"  >已有账号，立即登录</view>
        </navigator>
        <u-toast ref='uToast'/>
    </view>
</template>
<script>
    export default {
        data() {
            return{
                flag:true,
                phoneNum:'',
                nickName:'',
                password:''
            }
        },
        methods:{
            isSame(){

                uni.request({
                    url:"http://localhost:3004/users",
                    data:{
                        phoneNum:this.phoneNum
                    },
                    success:(res) => {
                        if(res.data.length != 0){
                            this.flag = false;
                            uni.showToast({
                                title:'手机号已注册!! ',
                                icon:'error'
```

```
                    })
                )
            )
        });
        console.log(this.flag);
        return this.flag;
    },
    submitReg() {
        if (this.phoneNum.length == 0) { //判断是否输入手机号
            //提示
            uni.showToast({
                title:'请先输入手机号！',
                icon:'error'
            })
        } else if (this.phoneNum.length != 11) {
            uni.showToast({
                title:'请输入正确手机号！',
                icon:'error'
            })
        } else if (this.nickName.length == 0) { //判断是否输入昵称
            uni.showToast({
                title:'请输入昵称！',
                icon:'error'
            })
        } else if (this.password.length == 0) { //判断是否输入密码
            uni.showToast({
                title:'请输入密码！',
                icon:'error'
            })
        } else if(this.isSame()) {
            //请求
            let URL = this.$baseUrl.baseUrl+'/users';
            uni.request({
                url:URL,
                data:{
                    phoneNum:this.phoneNum,
                    nickName:this.nickName,
                    password:this.password,
        avatar:"/image/avatar/" + Math.ceil(Math.random()*10) + '.jpeg',
                    typeid:1
                },
                method:'POST',
                header:{
                    'content-type':'application/x-www-form-urlencoded',
                },
                success:(res)=>{
                    this.$refs.uToast.show({
                        title:'登录成功',
                        type:'success'
                    }),
```

```
                        this.autoLogin()    ;
                    ),
                    fail:function(res) {
                        console.log("失败",res)
                    }
                ));
            )
        ),
        autoLogin(){
            uni.request({
                url:this.$baseUrl.baseUrl+'/users',
                method:"GET",
                data:{
                    phoneNum:this.phoneNum,
                    password:this.password,
                    typeId:this.typeid
                },
                success:(res) => {
                    console.log(res);
                    if (res.statusCode == 200) {
                        uni.setStorageSync('info', res.data[0]);
                        uni.switchTab({
                            url:'/pages/me/me'
                        });
                    }
                }
            });
        }
    }
    )
</script>
<style scoped>
    .registerWrap{
        width:100%;
        box-sizing:border-box;
        padding:0 40rpx;
    }
    .title { padding-top:60rpx;    }
    .title h3{
        font-size:24px;
        line-height:40px;
        color:#333;
    }
    .title h4{
        font-size:18px;
        color:#888;
    }
    .formWrap {    margin-top:20px; }
    .formItem{
        padding-bottom:15px;
    }
    .formItem .label{
        font-size:14px;
```

```
        color:#333;
    }
    .inputBox{
        margin-top:5px;
        border:1px #c3c3c3 solid;
        border-radius:8px;
    }
    .inputBox input{
        height:38px;
        line-height:38px;
        text-indent:1em;
    }
    .registTips{
        padding-top:20px;
        text-align:center;
    }
</style>
```

【代码解析】

（1）随机图像

在提交之前，使用 JS 代码进行验证，当验证通过后进行注册，注册的数据头像 avatar 的值为：'/image/avatar/' + Math.ceil(Math.random()*14) + '.jpeg'。在服务器的 public/image/avatar 目录中存放了 14 张 JPEG 图片。使用“Math.ceil(Math.random()*14)”随机得到 1～14 的整数。

（2）自动登录、注册方法

autoLogin 方法：实现自动登录功能，若登录成功，则调用“uni.setStorageSync('info', res.data[0]);”将用户信息存入缓存，并跳转到个人中心页。

submitReg 方法：实现注册功能，若注册成功，则调用 autoLogin 进行自动登录，并跳转到个人中心页。

（3）轻量弹框

注册页使用了两种轻量弹框，一种是 uni-upp 的内置组件 Toast，另一种是 uView 中的 u-Toast 组件。在页面的最后放置了一个轻量弹框“<u-toast ref='uToast' />”，在 JS 中通过“this.$refs.uToast”得到该组件实例。

2. 登录页

要使用下单功能、查看订单状态等都需要处于登录状态。登录成功，则将用户信息存入缓存。登录页效果如图 7-8 所示。页面完整的代码如下。

```
<template>
    <view class="loginWrap">
        <view class="logo">
            <image src="../../static/logo.png" mode="widthFix"></image>
        </view>
        <view class="title">
            <h3>登录</h3>
            <h4>欢迎使用智慧环保</h4>
        </view>
        <view class="formWrap">
            <view style="margin-bottom:10rpx;">
```

```
                <u-radio-group v-model="roleVal" placement="row">
                    <u-radio  label="普通用户" name="普通用户" :customStyle=
"{marginRight:'16px'}"  activeColor="rgb(0,194,151)"></u-radio>
                    <u-radio  label="企业用户" name="企业用户" activeColor="rgb
(0,194,151)"></u-radio>
                </u-radio-group>
            </view>
            <view class="formItem">
                <view class="label">手机号:</view>
                <view class="inputBox">
                    <input type="number" v-model="phoneNum" />
                </view>
            </view>
            <view class="formItem">
                <view class="label">密码:</view>
                <view class="inputBox">
                    <input type="number" password v-model="password" />
                </view>
            </view>
        </view>
        <view>
            <u-button type="primary" @click="loginSubmit" color="rgb(0,194,
151)"  >登 录</u-button>
        </view>
        <navigator url="register1">
            <view class="registTips" @tap="gotoReg">注册新用户</view>
        </navigator>
        <u-toast ref='uToast' />
    </view>
</template>
<script>
    export default {
        data() {
            return {
                phoneNum:'',
                password:'',
                roleVal:'普通用户',
                coverUrl:''
            }
        },
        methods:{
            gotoReg() {
                uni.navigateTo({
                    url:"/pages/login/register"
                })
            },
            loginSubmit() {
                if (this.phoneNum.length == 0) { //判断是否输入手机号
                    //提示
                    uni.showToast({
                        title:'请先输入手机号！',
```

```
                    icon:'error'
                })
            } else if (this.phoneNum.length != 11) {
                uni.showToast({
                    title:'请输入正确手机号！',
                    icon:'error'
                })
            } else if (this.password.length == 0) { //判断是否输入密码
                uni.showToast({
                    title:'请输入密码！',
                    icon:'error'
                })
            } else {
                //请求
                uni.request({
                    url:this.$baseUrl.baseUrl+'/users',
                    method:"GET",
                    data:{
                        phoneNum:this.phoneNum,
                        password:this.password,
                        typeid:this.roleVal == '普通用户' ? 1 :2
                    },
                    success:(res) => {
                        if (res.data.length > 0) {
                            console.log("成功:", res);
                            uni.setStorageSync('info', res.data[0]);
                            if (this.roleVal == '普通用户') {
                                this.$refs.uToast.show({
                                    message:'登录成功',
                                    type:'success'
                                })
                                );
                                uni.switchTab({
                                    url:'/pages/me/me'
                                })
                            } else {
                                this.$refs.uToast.show({
                                    message:'无效账号',
                                    type:'error'
                                })
                            }
                        }
                    });
                }
            )
        }
    }
</script>
<style>
    .loginWrap {
        width:100%;
```

```
        box-sizing:border-box;
        padding:0 40rpx;
    }
    .logo {
        width:30%;
        margin:0 auto;
    }
    .logo image {       width:100%;    }
    .title {       padding-top:60rpx;    }
    .title h3 {
        font-size:24px;
        line-height:40px;
        color:#333333;
    }
    .title h4 {
        font-size:18px;
        color:#888888;
    }
    .formWrap {
        padding-top:20rpx;
    }
    .formItem {
        padding-bottom:15px;
    }
    .formItem .label {
        font-size:14px;
        color:#333333;
    }
    .formItem .inputBox {
        border:1px #C3C3C3 solid;
        margin-top:5rpx;
        border-radius:8px;
    }
    .formItem .inputBox input {
        height:38px;
        line-height:38px;
        text-indent:1em;
    }
    .registTips {
        padding-top:20px;
        text-align:right;
        color:sandybrown;
    }
</style>
```

【代码解析】

单选按钮使用了 uView 中的组件 u-radio，u-radio 需配合 u-radio-group 使用。customStyle 属性可以自定义单选按钮的样式。u-radio-group 双向绑定 roleVal，单选按钮被选中，则值为 label 属性的值。

```
<u-radio-group v-model="roleVal" placement="row">
    <u-radio  label="普通用户" name="普通用户" :customStyle="{marginRight:'16px'}"
activeColor="rgb(0,194,151)"></u-radio>
```

```
        <u-radio  label="企业用户" name="企业用户" activeColor="rgb(0,194,151)">
</u-radio>
    </u-radio-group>
```

3. 个人中心页

个人中心页为导航栏中“我的”页面，页面效果如图 7-4 所示。在该页面中点击用户头像，会跳转到登录页。页面完整代码如下。

```
    <template>
        <view class="userCenter">
            <view class="userCard">
                <view class=" user-box  ">
                    <view class="user-img" @click="changeUser">
                        <u-avatar :src="pic" size="80"></u-avatar>
                    </view>
                    <view class="user-info">
                        <view class=" ">{{userName}}</view>
                        <view class="user-info-phone">{{phone}}</view>
                    </view>
                    <view class="sign">
                        <text>签到</text>
                    </view>
                    <view class="sign-icon">
                        <u-icon name="arrow-right" color="#fff" size="20"></u-icon>
                    </view>
                </view>
                <view class="overview">
                    <view class="view-item">
                        <view>{{revenue}}</view>
                        <view>累计收益</view>
                    </view>
                    <view class="view-item">
                        <view>{{orderTimes}}</view>
                        <view>回收次数</view>
                    </view>
                    <view class="view-item">
                        <view>{{totalScore}}</view>
                        <view>积分</view>
                    </view>
                </view>
            </view>
            <view class="overview ">
                <u-cell-group>
                    <u-cell icon="star" title="积分记录" isLink  :iconStyle="iconStyle">
</u-cell>
                    <u-cell icon="photo" title="兑换记录" isLink :iconStyle="iconStyle">
</u-cell>
                    <u-cell icon="coupon" title="我的贡献" isLink :iconStyle="iconStyle">
</u-cell>
                    <u-cell icon="heart" title="收入记录" isLink :iconStyle="iconStyle">
</u-cell>
                    <u-cell icon="setting" title="设置" isLink :iconStyle="iconStyle">
</u-cell>
```

```
            </u-cell-group>
        </view>
    </view>
</template>
<script>
    export default {
        data() {
            return {
                pic:'../../static/avatar.png',
                show:true,
                userName:'小甜鱼',
                phone:'',
                orderTimes:23,
                revenue:100,
                totalScore:100,
                iconStyle:{
                    padding:'8px 8px'
                }
            }
        },
        onShow() {
            this.show = uni.getStorageSync('info') != '';
            if(this.show){
                this.userName = uni.getStorageSync('info').nickName;
                this.phone = uni.getStorageSync('info').phoneNum.substr
(0,7)+'****';
                console.log(uni.getStorageSync('info').avatar);
                this.pic = uni.getStorageSync('info').avatar != '' ? this.$baseUrl.
baseUrl + uni.getStorageSync('info').avatar :'https://uviewui.com/common/logo.png';
            }else{
                this.userName ="未登录",
                this.phone = '';
                this.orderTimes=0;
                this.revenue=0;
                this.totalScore= 0;
            }
        },
        methods:{
            changeUser(){
                uni.navigateTo({
                    url:'/pages/login/login'
                })
            }
            //如果有积分记录、回收次数等功能的需求，可以添加相应代码，这里省略此部分代码
        }
    }
</script>
<style lang="scss">
page{   background-color:#ededed; }
.user-box{
    padding-top:10px;
    color:#fff;
    display:flex;
```

```
        padding:30 10 20 30;
        align-items:center;
    }
    .userCenter{
        box-sizing:border-box;
        padding:0 15px;
        background-color:#fff;
        overflow:hidden;
    }
    .userCard{
        margin:15px auto ;
        background-color:#71d9ca;
        border-radius:10px;
        padding:10px;
        margin-bottom:0px;
    }
    .user-info{
        flex:1;
        font-size:18px;
        margin-left:15px;
    }
    .user-info-phone{    font-size:15px;  }
    .sign{
        font-size:15px;
        margin-right:15px;
    }
    .overview{
        display:flex;
        align-items:center;
        padding-top:15px;
    }
    .view-item{
        flex:1;
        font-size:16px;
        text-align:center;
        color:#fff;
        line-height:24px;
    }
    .userDetail{
        padding-top:20px;
        padding-bottom:20px;
    }
    .iconStyle{    padding-top:50px;  }
    </style>
```

【代码解析】

页面中的列表使用了 uView 的 u-cel 组件，该组件需要搭配 u-cell-group 使用，用 u-cell-group 实现列表的上下边框。这里 icon 表示列表项左侧的图标，title 表示列表项左侧的标题，默认情况下，列表项右侧有箭头。iconStyle、righticonStyle、titleStyle 属性分别设置列表项左侧的图标、右侧箭头、左侧标题的样式。更多的样式读者可以参考 uView 官网的介绍。

```
    <u-cell-group>
        <u-cell icon="star" title="积分记录" isLink  :iconStyle="iconStyle"></u-cell>
    </u-cell-group>
```

7.3.5 公司回收页、公司详情页、公司搜索结果页

1. 公司回收页

公司回收页展示回收公司的信息，可以由此页面进入下单页，页面效果如图 7-3 所示。公司信息数据来源地址：http://localhost:3004/companys。点击公司列表项进入公司详情页。另外，有可能公司比较多，在公司信息列表外面加一个 sroll-view 容器，当公司信息在一个页面显示不下时，可以往下滑动。

完整的代码如下。

```
<template>
    <view id="companyRecycle">
        <view class="cpHead">
            <view class="title">
                <image mode="widthFix" src="../../static/app_logo.png"></image>
                <text>公司回收</text>
            </view>
            <view class="search">
                <u-search placeholder="请输入搜索内容" height="50" shape="square"
v-model="keyword" @search="search"  :showAction="false" ></u-search>
            </view>
        </view>
        <view class="container">
            <scroll-view scroll-y="true" class="companylist">
                <view class="company-item" v-for="item in cpnlist">
                    <view class="item-title">{{item.name}}</view>
                    <view class="item-cont"  @click="gotoDetail(item.id)" >
                        <view class="item-img">
                            <image :src="$baseUrl.baseUrl+item.coverUrl"></image>
                        </view>
                        <view class="cont-right">
                            <view class="right-info">{{item.address}}</view>
                            <view class="right-info">营业时间:8:00—17:00</view>
                            <view class="right-info">联系电话:{{item.contact}}</view>
                        </view>
                    </view>
                    <view class="placeOrder">
    <u-button @click="gotoFillIn(item.id)" type="primary" color="rgb(0,194,151)">
立即下单</u-button>
                    </view>
                </view>
            </scroll-view>
        </view>
    </view>
</template>
<script>
    export default {
        data() {
            return {
                cpnlist:[],
```

```
                keyword:''
            }
        },
        onLoad() {       this.getCompantlist();    },
        methods:{
            getCompantlist() {
                uni.$u.http.get(this.$baseUrl.baseUrl+'/companys', {}).then
(res => {
                console.log(res);
                if (res.statusCode == 200) {
                    this.cpnlist = res.data;
                    console.log(this.cpnlist);
                }
            })
            },
            gotoDetail(id) {
                uni.navigateTo({
                    url:'/childPages/company/detail?id=' + id
                })
            },
            search(e) {
                uni.navigateTo({
                    url:'/childPages/company/searchRes?text=' + e
                })
            },
            gotoFillIn(id,name) {
                uni.navigateTo({
    url:'/childPages/company/fillIn?companyId=' + id+'&companyName='+name
                })
            }
        }
    }
    </script>
    <style scoped>
        #companyRecycle {       background-color:#F7fafb;    }
        .cpHead {
            box-sizing:border-box;
            padding:20px;
            background-color:rgb(0, 194, 151);
            height:200px;
        }
        .cpHead .title {
            font-size:20px;
            color:#fff;
            display:flex;
            align-items:center;
        }
        .cpHead .title image {    width:60px;    }
        .search {
            width:100%;
            box-sizing:border-box;
            padding:30px 20px 15px 20px;
            background-color:rgb(0, 194, 151);
        }
```

```
    .container {
        width:100%;
        height:calc(100vh - 250px);
        overflow:hidden;
    }
    .companylist {
        box-sizing:border-box;
        padding:0 20px;
        height:100%;
    }
    .company-item {
        box-sizing:border-box;
        padding:10px;
        margin:15px  5px 0 5px;
        background-color:#fff;
        border-radius:8px;
        box-shadow:0 1px 8px rgba(14, 197, 156, 0.4);
    }
    .company-item::after {
        content:'';
        display:block;
        clear:both;
    }
    .item-title {
        font-size:18px;
        border-bottom:1rpx #e6e6e6 solid;
        line-height:36px;
    }
    .item-cont {
        display:flex;
        padding:5px 0;
    }
    .item-img {
        flex:.8;
    }
    .item-img image {
        width:100%;
        height:90px;
    }
    .cont-right {
        flex:2;
        margin-left:10px;
        display:flex;
        flex-flow:column;
        justify-content:space-between;
    }
    .placeOrder {
        width:30%;
        float:right;
    }
</style>
```

2. 公司详情页

公司详情页展示单个公司的详细信息，页面效果如图 7-11 所示。其 API 地址为：

http://localhost:3004/companys?id=id。完整的代码如下。

```
    <template>
        <view>
            <view class="cpHead">
                <view class="company-item">
                    <view class="item-title">{{companyDetail.name}}</view>
                    <view class="item-cont">
                        <view class="item-img">
                <image :src="$baseUrl.baseUrl + companyDetail.coverUrl"></image>
                        </view>
                        <view class="cont-right">
                            <view class="right-info">地址:{{companyDetail.address}}
</view>
            <view class="right-info">联系电话:{{companyDetail.contact}}</view>
                            <view class="right-info">评价等级:{{companyDetail.score}}
</view>
                        </view>
                    </view>
                </view>
            </view>
            <view class="introduce">{{companyDetail.introduction}}</view>
            <view class="appointBtn">
    <u-button @click="gotoFillIn(companyDetail.id)"  color="rgb(0,194,151)">预约
上门回收</u-button>
            </view>
        </view>
    </template>
    <script>
        export default {
            data() {
                return {
                    companyDetail:{}
                }
            },
            onLoad(options) {
                this.getCompanyDtl(options.id)
            },
            methods:{
                getCompanyDtl(id) {
                    uni.$u.http.get(this.$baseUrl.baseUrl + '/companys?id='+id,{}).then
( res => {
                        console.log(res);
                            if(res.statusCode == 200) {
                                this.companyDetail = res.data[0];
                                console.log(this.companyDetail);
                            }
                    })
                },
                gotoFillIn(id) {
                    uni.navigateTo({
                        url:'/childPages/company/fillIn?id='+id
                    })
```

```
            )
        )
    )
</script>
<style scoped>
    .cpHead{
        box-sizing:border-box;
        padding:20px;
        background-color:rgb(0,194,151);
    }
    .item-title{
        font-size:18px;
        line-height:36px;
        color:#fff;
    }
    .item-cont{
        display:flex;
        padding:5px 0;
        margin-top:10px;
    }
    .item-img{
        flex:.8;
    }
    .item-img image{
        width:100%;
        height:90px;
    }
    .cont-right{
        flex:2;
        margin-left:10px;
        display:flex;
        flex-flow:column;
        justify-content:space-between;
        color:#fff;
    }
    .introduce{
        box-sizing:border-box;
        padding:20px;
    }
    .appointBtn{
        width:calc(100% - 40px);
        position:fixed;
        left:20px;
        bottom:20px;
    }
</style>
```

3. 公司搜索结果页

公司回收页、公司搜索结果页中的搜索框用于输入搜索关键字来搜索名称含有关键字的公司，并将搜索结果显示在公司搜索结果页中，页面效果如图 7-10 所示。其 API 地址为：http://localhsot:3004/companys?name_like=公司名称。这里的公司搜索结果页与分类查询结果页类似，不做过多解释，页面完整代码如下。

```
<template>
    <view>
        <view class="search">
            <u-search placeholder="请输入搜索内容" shape="square" v-model=
"keyword" @search="search" actionText="搜索"      :showAction="true"></u-search>
        </view>
        <view class="container">
            <scroll-view scroll-y="true" class="companylist" v-if="cpnlist.
length > 0">
                <view class="company-item" v-for="(item,index) in cpnlist"
:key="index" @click="gotoDetail(item.id)">
                    <view class="item-title">{{item.name}}</view>
                    <view class="item-cont">
                        <view class="item-img">
                            <image :src="$baseUrl.baseUrl + item.coverUrl"></image>
                        </view>
                        <view class="cont-right">
                            <view class="right-info">{{item.address}}</view>
                            <view class="right-info">营业时间:8:00—17:00</view>
                            <view class="right-info">联系电话:{{item.contact}}</view>
                        </view>
                    </view>
                    <view class="placeOrder">
                        <u-button :customStyle="customStyle" type="primary" color=
"rgb(0,194,151)">立即下单</u-button>
                    </view>
                </view>
            </scroll-view>
            <view class=" emptyBox" v-else>
                <u-empty text="暂无搜索数据!" mode="data"></u-empty>
            </view>
        </view>
    </view>
</template>
<script>
    export default {
        data() {
            return {
                cpnlist:[],
                keyword:''
            }
        },
        onLoad(options) {
            this.keyword = options.text;
            this.getSearchRes(options.text)
        },
        methods:{
            gotoDetail(id) {
                uni.navigateTo({
                    url:'/childPages/company/detail?id=' + id
                })
            },
            search(e) {
```

```
                this.getSearchRes(e);
            },
            getSearchRes(str) {
                uni.$u.http.get(this.$baseUrl.baseUrl + "/companys?name_like="
+ str, {

                }).then(res => {
                    console.log(res);
                    if (res.statusCode == 200) {
                        this.cpnlist = res.data;
                        console.log(this.cpnlist);
                    }
                })
            }
        }
    }
</script>
<style scoped>
    .customStyle {
        backgroundColor:'rgb(0,194,151)';
        color:'white';
        height:'32px';
        fontSize:'14px';
        borderRadius:'16px'
    }
    .result {   background-color:#f7fafd;   }
    .search {
        width:100%;
        box-sizing:border-box;
        padding:15px 20px 15px 20px;
        background-color:rgb(0, 194, 151);
    }
    .container {
        width:100%;
        height:600px;
        height:calc(100vh - 130px);
        overflow:hidden;
    }
    .companylist {
        box-sizing:border-box;
        padding:0 15px;
        height:100%;
    }
    .company-item {
        box-sizing:border-box;
        padding:10px;
        margin:15px 5px 0 5px;
        background-color:#fff;
        border-radius:8px;
        box-shadow:0 1px 8px rgba(14, 197, 156, 0.4);
    }
    .company-item::after {
        content:'';
        display:block;
```

```
            clear:both;
        }
        .item-title {
            font-size:18px;
            border-bottom:1rpx #e6e6e6 solid;
            line-height:36px;
        }
        .item-cont {
            display:flex;
            padding:5px 0;
        }
        .item-img {    flex:.8;    }
        .item-img image {
            width:100%;
            height:90px;
        }
        .cont-right {
            flex:2;
            margin-left:10px;
            display:flex;
            flex-flow:column;
            justify-content:space-between;
        }
        .placeOrder {
            width:30%;
            margin-top:5px;
            float:right;
        }
        .emptyBox {
            position:fixed;
            top:50%;
            left:50%;
            transform:translate3d(-50%, -50%, 0);
        }
    </style>
```

7.3.6 下单页、订单页、订单详情页

1. 下单页

从公司相关的页面跳转至下单页，跳转时，url 会携带公司的“id”、名称，下单的操作结果就是在订单列表中增加一条记录。页面效果如图 7-9 所示。使用 POST 请求的方式访问 API 地址 http://localhost:3004/companys。由于后端服务器为本地模拟服务器，上传图片不被支持，故此处订单图片为指定图片，上传图片的功能省略。页面完整代码如下。

```
    <template>
        <view class="formWrap">
            <u-form :model="form" ref="uForm" label-width="80" labelPosition=
"left" :rules="rules">
                <u-form-item label="物品类别" prop="order.type" borderBottom ref="type"
@click="typeShow = true">
    <u-input v-model="form.order.type" placeholder="请选择物品类别" border="none">
</u-input>
```

```
                <u-icon slot="right" name="arrow-right"></u-icon>
                <u-picker :show="typeShow" @confirm="typeCfm" @change=
"changeHandler" @cancel="typeCancel"
                    :columns="typelist" ref="uPicker" v-model="typeShow">
</u-picker>
            </u-form-item>
            <u-form-item label="联系人" prop="order.name" borderBottom>
    <u-input v-model="form.order.name" placeholder="请输入" border="none">
</u-input>
            </u-form-item>
            <u-form-item label="联系方式" prop="order.phone" borderBottom>
    <u-input v-model="form.order.phone" placeholder="请输入" border="none">
</u-input>
            </u-form-item>
            <u-form-item label="上门地址" prop="order.address" borderBottom>
    <u-input v-model="form.order.address" placeholder="请输入" border="none">
</u-input>
            </u-form-item>
    <u-form-item label="上门时间" prop="order.time" borderBottom @click="timeShow
= true">
    <u-input v-model="form.order.time" placeholder="请选择上门时间" border="none"
disabled disabledColor="#ffffff">       </u-input>
                <u-icon slot="right" name="arrow-right"></u-icon>
                <u-action-sheet :show="timeShow" :actions="timelist" title="请
选择时间" description="选择上门时间"
                    @close="timeShow = false" @select="timeCfm">
                </u-action-sheet>
            </u-form-item>
            <u-form-item label="物品重量" prop="order.weight" borderBottom>
    <u-input v-model="form.order.weight" placeholder="请输入" border="none">
</u-input>
            </u-form-item>
            <u-form-item label="物品件数" prop="order.number" borderBottom>
    <u-input v-model="form.order.number" placeholder="请输入" border="none">
</u-input>
            </u-form-item>
        </u-form>
        <view style="margin:15px auto;">
            <u-upload :fileList="fileList" @afterRead="afterRead" @delete=
"deletePic" name="2" multiple :maxCount="10">
            </u-upload>
        </view>
        <view class="details">
            <textarea v-model="content" placeholder="请输入其他信息"></textarea>
        </view>
        <view class="appointBtn">
    <u-button type="primary" @click="appointment" color="rgb(0,194,151)">立即预
约</u-button>
        </view>
        <u-toast ref="uToast"></u-toast>
    </view>
```

```
</template>
<script>
    oxport default {
        data() {
            return {
                fileList:[],
                latitude:39.909,
                longitude:116.39742,
                form:{
                    order:{
                        type:'',
                        name:'',
                        phone:'',
                        address:'',
                        time:'',
                        weight:'',
                        number:''
                    }
                },
                rules:{
                    'order.type':[{
                        required:true,
                        message:'请选择物品类别',
                        trigger:['change']
                    }],
                    'order.name':[{
                        required:true,
                        message:'请输入联系人',
                        trigger:['blur']
                    }],
                    'order.phone':[{
                        required:true,
                        type:'number',
                        message:'请输入联系方式',
                        trigger:['blur']
                    }],
                    'order.address':[{
                        required:true,
                        message:'请输入地址',
                        trigger:['blur']
                    }],
                    'order.time':[{
                        required:true,
                        message:'请选择上门时间',
                        trigger:['change']
                    }],
                    'order.weight':[{
                        required:true,
                        type:'float',
                        message:'请输入物品重量',
                        trigger:['blur']
                    }],
```

```
                'order.number':[{
                    required:true,
                    message:'请输入物品件数',
                    type:'integer',
                    trigger:['blur']
                }],
            },
            typeShow:false,
            timeShow:false,
            typelist:[
                ['废纸', '塑料', '金属', '纺织品', '家电'],
                ['报纸', '书籍', '纸箱']
            ],
            typeData:[
                ['报纸', '书籍', '纸箱'],
                ['水瓶', '桶', '塑料袋', '农用地膜', '泡沫塑料', '塑料编织品'],
                ['铜丝', '铁制品', '废钢筋', '不锈钢', '废银锌电池'],
                ['羽绒服', '毛衣', '棉絮', '上衣', '裤子', '裙子'],
                ['电视机', '空调', '冰箱', '洗衣机', '笔记本电脑', '微波炉', '烤箱']
            ],
            typeId:null,
            unitPrice:null,
            timelist:[{
                name:'8:00—10:00'
            }, {
                name:'10:00—12:00'
            }, {
                name:'14:00—16:00'
            }, {
                name:'16:00—18:00'
            }],
            photoUrl:'',
            content:'',
            companyId:null
        }
    },
    onReady() {
        this.$refs.uForm.setRules(this.rules);
    },
    onLoad(options) {
        let user = uni.getStorageSync('info');
        if (user == '') {
            uni.navigateTo({
                url:'/pages/login/login'
            })
        }
        this.companyId = options.id;
        this.companyName = options.Name;
    },
    methods:{
```

```
            changeHandler(e) {
                const {
                    columnIndex,
                    value,
                    values, // values为当前变化列的数组内容
                    index,
                    // 微信小程序无法将picker实例传出来，只能通过ref操作
                    picker = this.$refs.uPicker
                } = e
                // 当第一列值发生变化时，变化第二列（后一列）对应的选项
                if (columnIndex == 0) {
                    // picker为选择器this实例，变化第二列对应的选项
                    picker.setColumnValues(1, this.typeData[index])
                }
            },
            typeCancel() {
                this.typeShow = false;
            },
            typeCfm(e) {
                console.log(e);
                console.log(e.value[1]);
                this.form.order.type = e.value[1];
                this.typeShow = false;
            },
            timeCfm(e) {
                console.log(e.name);
                this.form.order.time = e.name;
            },
            appointment() {

                let userId = uni.getStorageSync('info').id;
                this.photoUrl = '';
                this.unitPrice = 2.3;
                this.$refs.uForm.validate().then(valid => {
                    console.log("valid=" + valid);
                    if (valid) {
                        uni.request({
                            url:this.$baseUrl.baseUrl + '/orders',
                            method:'POST',
                            data:{
                                "companyName":this.companyName,
                                "createby":this.form.order.name, //联系人
                                "createTime":this.$u.timeFormat(new Date(), "yyyy-
mm-dd hh:MM:ss"),
                                "pickupTime":this.form.order.time, //上门时间
                                "pickupAddress":this.form.order.address, //上门地址
                                "goodNumber":this.form.order.weight, //物品重量
                                "goodUnitPrice":this.unitPrice, //回收物品单价
                                "goodTypeName":this.form.order.type, //回收类型
                                "coverUrl":"/image/order/paper.jpeg",
                                "contact":this.form.order.phone, //联系方式
```

```
                        "remark":this.content, //备注
                        "pkgs":this.form.order.pkgs, //物品包裹数量
                        "state":1,
                        "companyId":this.companyId,
                        "userId":userId
                    },
                    success:(res) => {
                        console.log(res);
                        if (res.statusCode == 201) {
                            this.$refs.uToast.show({
                                message:'预约成功',
                                type:'success',
                                isTab:true

                            })
                            uni.switchTab({
                                url:'/pages/order/order'
                            })
                        } else {
                            uni.showToast({
                                message:'预约失败，请重试！',
                                icon:'error'
                            })
                        }
                    }
                })
            }
        })
    }
}
</script>
<style scoped>
    .formWrap {
        box-sizing:border-box;
        padding:0 15px 70px 15px;
    }
    .details {
        border:1rpx #E6E6E6 solid;
        border-radius:5px;
    }
    .appointBtn {
        width:calc(100% - 40px);
        position:fixed;
        left:20px;
        bottom:20px;
    }
</style>
```

【代码解析】

该页面使用了 uView 的表单组件 u-form。在<u-form></u-form>标签内通过<u-form-item></u-form-item>标签添加表单项，<u-form-item></u-form-item>标签内除了 input 外，还可以放

置其他的元素。本页面中应用的 u-form 组件的属性及其含义如下。

- model：表单数据对象。
- labelWidth：标签的宽度，默认单位为 px。
- labelPosition：标签的位置，可选值有 top、left，默认值为 left。
- rules：表单规则，这里对应的是一个规则对象。

本页面中应用的 u-form-item 组件的属性及其含义如下。

- label：表单项的文本标签。
- borderBottom：是否显示表单项的底部边框。
- prop：表单数据对象的属性名，在使用 validate、resetFields 方法的情况下，该属性是必填的。本页面使用了 validate 方法，对表单数据进行有效性验证。下面是联系方式的规则定义，该字段的值只能为数字，不能为空，失去焦点时进行验证，如验证失败，则在下方显示“请输入联系方式”。表单验证的效果如图 7-18 所示。

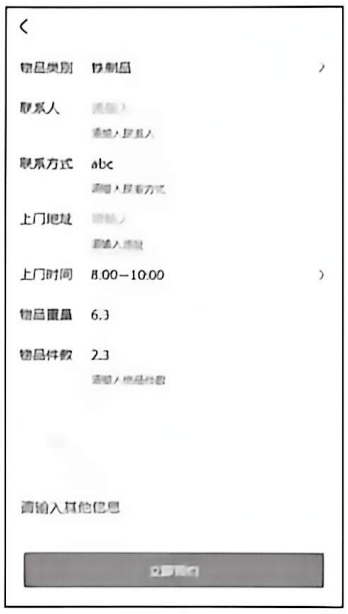


图 7-18　表单验证效果

```
rules:{
……省略
        'order.phone':[{
            required:true,
            type:'number',
            message:'请输入联系方式',
            trigger:['blur']
        }],
            ……省略
}
```

在 JS 代码中通过下列方法来判断是否通过验证，若 valid 的值为 true，则表示通过验证。

```
this.$refs.uForm.validate().then(valid => {
if (valid) {  }
}
```

本页面中将 input（单行输入框）的边框去掉，然后显示表单项的底部边框。在设置【物品类别】【上门时间】这两项时都会在页面底部弹出选择器或者操作菜单进行选择。效果如图 7-19 和图 7-20 所示。这里【物品类别】选择的是 uView 的 u-picker 组件，实现多级关联选择，【上门时间】选择的是 u-Action-Sheet 操作菜单组件。

2. 订单页

订单页展示当前用户的待接单、已接单、已完成、已取消等多种状态的订单信息，如图 7-2 所示。下面只列举当前用户的待接单信息，通过 GET 请求方式访问 API 获取数据。

API 地址为 http://localhost:3004/orders?userId=用户 id &state=1。

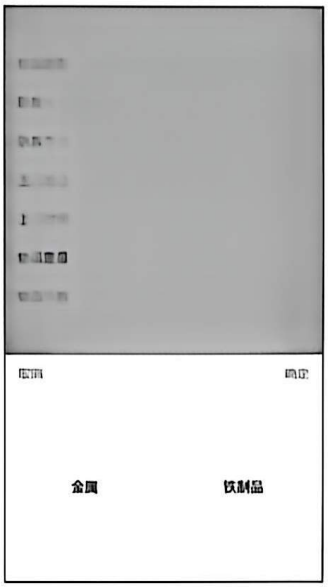


图 7-19　多级关联选择

图 7-20　操作菜单选择

其他的订单信息显示方法与此类似，这里不做讲解。以下只列出待接单信息显示的代码，具体代码如下。

```
    <template>
        <view class="wrap">
            <view class="u-tab-box">
            <u-subsection :list="list" :current="swiperCurrent" @change=
"sectionChange" activeColor="#00c297"  bgColor="red"
                mode="subsection"></u-subsection>
           </view>
            <swiper class="swiper-box" :current="swiperCurrent" >
                <swiper-item class="swiper-item">
                    <scroll-view scroll-y style="height:100%;width:100%;" >
                        <view class="page-box">
        <view class="order" v-for="(item,index) in dataList1"  :key="item.id"
                            @click="gotoDetail(item.id)">
                            <view class="top">
                                <view class="left">
                                    <view class="store"> 已等待:{{$u.timeFrom
(getUnixTime(item.createTime+''),'yyyy-mm-dd')}}</view>
                                </view>
                                <view class="right">待接单</view>
                            </view>
                            <view class="item">
                                <view class="left">
                    <image :src="$baseUrl.baseUrl+item.coverUrl" mode="aspectFill">
</image>
                                </view>
                                <view class="content">
            <view class="title u-line-2">回收的类型:{{ item.goodTypeName }}</view>
            <view class="type">重量:{{ item.goodNumber}}kg</view>
            <view class="type">预计收入金额:{{ item.goodNumber*item.goodUnitPrice}}
元</view>
```

```
        <view class="type">取货方式:上门收货(( item.pickupTime))</view>
                            </view>
                        </view>
                        <view class="bottom">
                            <view class="more">
                <u-icon name="more-dot-fill" color="rgb(203,203,203)"></u-icon>
                            </view>
            <view class="evaluate btn" @tap="cancelOrder(item.id)">取消订单</view>
                                </view>
                    </view>
                </view>
            </scroll-view>
        </swiper-item>
        <swiper-item>已接单订单列表</swiper-item>
        <swiper-item>已完成订单列表</swiper-item>
        <swiper-item>已取消订单列表</swiper-item>
    </swiper>
    <u-toast ref='uToast' ></u-toast>/>
</view>
</template>
<script>
    export default {
        data() {
            return {
                list:['待接单', '已接单', '已完成', "已取消"],
                swiperCurrent:0,
                dataList1:[]
            }
        },
        onLoad() {
            this.getOrders();
        },
        methods:{
            getUnixTime(dateStr) {
                let newstr = dateStr.replace(/-/g, '/');
                let date = new Date(newstr);
                let time_str = date.getTime().toString();
                return time_str.substr(0, 10);
            },
            sectionChange(index) {
                this.swiperCurrent = index;
            },
            getOrders(idx) {
                let user = uni.getStorageSync('info');
                if(user == ''){
                    uni.navigateTo({
                        url:'/pages/me/login'
                    })
                }
                let userId = user.id;
                let url = this.$baseUrl.baseUrl+'/orders?state=1&userId=' + userId;
                uni.request({
                    url:url,
```

```
                method:'GET',
                success:(res) => {
                    console.log(res);
                    this.dataList1 = res.data;
                }
            })
        },
        gotoDetail(id) {
            uni.navigateTo({
                url:'/childPages/order/waiting?id=' + id
            })
        },
        cancelOrder(id){
            console.log(id);
            uni.request({
                url:this.$baseUrl.baseUrl+'/orders/'+id,
                method:'patch',
                data:{
                    state:0
                },
                success:(res) => {
                    if(res.statusCode == 200){
                        this.$refs.uToast.show({
                            type:'success',
                            message:'修改成功，即将刷新页面',
                            duration:3000
                        })
                    };
                    location.reload();//刷新页面
                }
            })
        }
    }
}
</script>
<style lang="scss">
    .order {
        width:710rpx;
        background-color:#ffffff;
        margin:20rpx auto;
        border-radius:20rpx;
        box-sizing:border-box;
        padding:20rpx;
        font-size:28rpx;
        border-bottom:1px solid #dddddd;
        .top {
            display:flex;
            justify-content:space-between;
            .left {
                display:flex;
                align-items:center;
                .store {
                    margin:0 10rpx;
                    font-size:32rpx;
```

```
                font-weight:bold;
            }
        }
        .right {
            color:#00c297;
        }
    }
    .item {
        display:flex;
        margin:20rpx 0 0;

        .left {
            margin-right:20rpx;

            image {
                width:200rpx;
                height:200rpx;
                border-radius:10rpx;
            }
        }
        .content {
            .title {
                font-size:28rpx;
                line-height:50rpx;
            }
            .type {
                margin:10rpx 0;
                font-size:24rpx;
                color:$u-tips-color;
            }
            .delivery-time {
                color:#e5d001;
                font-size:24rpx;
            }
        }
        .right {
            margin-left:10rpx;
            padding-top:20rpx;
            text-align:right;

            .decimal {
                font-size:24rpx;
                margin-top:4rpx;
            }
            .number {
                color:$u-tips-color;
                font-size:24rpx;
            }
        }
    }
    .total {
        margin-top:20rpx;
        text-align:right;
        font-size:24rpx;
```

```
            .total-price {
                font-size:32rpx;
            }
        }
        .bottom {
            display:flex;
            margin-top:40rpx;
            padding:0 10rpx;
            justify-content:space-between;
            align-items:center;
            .btn {
                line-height:60rpx;
                width:160rpx;
                border-radius:26rpx;
                border:2rpx solid $u-border-color;
                font-size:26rpx;
                text-align:center;
                color:skyblue;
            }
            .evaluate {
                color:#00c297;
                border-color:#00c297;
            }
        }
    }
    .centre {
        text-align:center;
        margin:200rpx auto;
        font-size:32rpx;
        image {
            width:164rpx;
            height:164rpx;
            border-radius:50%;
            margin-bottom:20rpx;
        }
        .tips {
            font-size:24rpx;
            color:#999999;
            margin-top:20rpx;
        }
        .btn {
            margin:80rpx auto;
            width:200rpx;
            border-radius:32rpx;
            line-height:64rpx;
            color:#ffffff;
            font-size:26rpx;
            background:linear-gradient(270deg, rgba(249, 116, 90, 1) 0%,
rgba(255, 158, 1, 1) 100%);
        }
    }
    .wrap {
        display:flex;
        flex-direction:column;
```

```
            height:calc(100vh - var(--window-top));
            width:100%;
        }
    .u-tab-box{
        height:40px;
    }
        .swiper-box {
            flex:1;
        }
        .swiper-item {
            height:100%;
        }
    </style>
```

【代码解析】

（1）实现分段显示

分段显示使用了 uView 中的分段器 u-subsection 组件。当点击顶部分段器时，current 属性会发生对应变化，与 current 属性双向绑定的 JS 代码的 data 中的属性 swiperCurrent 会跟着变化。

```
        <u-subsection :list="list" :current="swiperCurrent"></u-subsection>
</template>
```

使用轮播组件显示每段的内容，轮播组件的 current 属性同样绑定属性 swiperCurrent。这样分段器和轮播的当前项可以同步。

（2）显示数据

订单信息可能一屏显示不完，所以需要使用 scroll-view 组件，达到可滚动查看的效果。点击订单项会跳到订单详情页 waiting.vue。这里只实现了待接单的订单详情页，其他状态的订单详情页与之类似，故不做介绍。

（3）取消订单

取消订单就是将订单的状态属性 state 值改为 1。json-server 服务器可使用 PATCH 请求方式，对数据做局部更新。订单取消后，需要刷新订单页，cancelOrder 方法实现了该功能。

3. 订单详情页

在订单页中点击订单信息，会跳转到订单详情页。页面效果如图 7-12 所示。该页面的布局与公司详情页的类似，主要实现取消订单的功能。页面完整的代码如下。

```
    <template>
        <view>
            <view class="top">
                <view class="left">      <view class="store">订单详情</view>    </view>
                <view class="right">待接单</view>
        </view>
        <view class="cpHead">
                <view class="company-item">
                    <view class="item-title">{{orderDetail.name}}</view>
                    <view class="item-cont">
                        <view class="item-img">
        <image :src="$baseUrl.baseUrl +orderDetail.coverUrl" mode="aspectFill">
</image>
                        </view>
                        <view class="cont-right">
```

```
        <view class="right-info">回收类型:{{orderDetail.goodTypeName}}</view>
        <view class="right-info">重量:{{orderDetail.goodNumber}}kg</view>
        <view class="right-info">预计收入金额:{{ orderDetail.goodNumber*
orderDetail.goodUnitPrice}}元</view>
                    </view>
                </view>
            </view>
        </view>
        <view class="content">
            <view>订单编号:{{orderId}}</view>
            <view>取货方式:上门取货</view>
            <view>取货时间:{{orderDetail.pickupTime}}</view>
            <view>取货地址:{{orderDetail.pickupAddress}}</view>
            <view>备注:{{orderDetail.remark!=null?orderDetail.remark:'暂无'}}
</view>
        </view>
        <view class="introduce">{{orderDetail.introduction}}</view>
        <view class="appointBtn">
            <u-button   @click="cancelOrder(orderDetail.id)"   type="primary"
color="rgb(0,194,151)"  >取消订单</u-button>
        </view>
        <u-toast ref="uToast"></u-toast>
    </view>
</template>
<script>
    export default {
        data() {
            return {
                orderId:'',
                orderDetail:{}
            }
        },
        onLoad(options) {
            this.getOrderDtl(options.id)
        },
        methods:{
            getUnixTime(dateStr) {
                let date = new Date(dateStr);
                let time_str = date.getTime().toString();
                return time_str.substr(0, 10);
            },
            getOrderDtl(id) {
                uni.$u.http.get(this.$baseUrl.baseUrl + '/orders?id='+ id ).then
(res => {
                    if (res.statusCode == 200) {
                        console.log(res);
                        this.orderDetail = res.data[0];
                        let str = this.getUnixTime(res.data[0].createTime);
                        this.orderId = str.toString() + id;
                    }
                })
            },
            cancelOrder(id){
```

```
                console.log(id);
                uni.request({
                    url:this.$baseUrl.baseUrl+'/orders/'+id,
                    method:'patch',
                    data:{
                        state:0
                    },
                    success:(res) => {
                        if(res.statusCode == 200){
                            this.$refs.uToast.show({
                                type:'success',
                                message:'修改成功，即将刷新页面',
                                duration:3000
                            });
                            uni.navigateBack();//回到上一级页面
                        }else{
                                this.$refs.show({
                                    message:'操作失败!! ',
                                    type:'error',

                                })
                        }

                    }
                })
            }
        }
    }
</script>
<style lang="scss" scoped>
    .top {
        display:flex;
        justify-content:space-between;
        margin-bottom:20rpx;
        margin-left:20rpx;
        margin-right:20rpx;
        .left {
            display:flex;
            align-items:center;
            border-left:4rpx solid rgb(0, 194, 151);
            .store {
                margin:0 10rpx;
                font-size:32rpx;
                font-weight:bold;
            }
        }
        .right {
            color:rgb(255, 100, 0);
            font-size:32rpx;
            font-weight:bold;
        }
    }
    .cpHead {
        box-sizing:border-box;
```

```
        padding:20px;
        background-color:rgb(0, 194, 151);
        margin:0 20rpx;
        border-radius:8rpx;
    }
    .item-title {
        font-size:18px;
        line-height:36px;
        color:#fff;
    }
    .item-cont {
        display:flex;
        padding:5px 0;
        margin-top:10px;
    }
    .item-img {    flex:.8;    }
    .item-img image {
        width:100%;
        height:90px;
    }
    .cont-right {
        flex:2;
        margin-left:10px;
        display:flex;
        flex-flow:column;
        justify-content:space-between;
        color:#fff;
    }
    .introduce {
        box-sizing:border-box;
        padding:20px;
    }
    .appointBtn {
        width:calc(100% - 40px);
        position:fixed;
        left:20px;
        bottom:20px;
    }
    .content {
        font-size:15px;
        line-height:30px;
        margin-top:20px;
        margin-left:20px;
    }
</style>
```

本章小结

本章基于 json-server 工具模拟后端数据接口，结合 uView 2.0 组件开发了一个智慧环保项目。json-server 是一个在前端本地运行，可以存储 JSON 数据的服务器。uView 是一个优秀的 uni-app 生态框架，其全面的组件和便捷的工具可以帮助用户快速开发移动应用项目。智

慧环保项目功能完善、链接正常，能培养读者的综合实践能力。

项目实战

由于篇幅的限制，智慧环保项目中的积分商城、附近回收、设置、公司端等功能尚未实现，请继续完善项目。表 7-3 列出了该项目的部分扩展功能，仅供参考。

表 7-3　项目的部分扩展功能

模块	功能	功能描述
积分商城	商品列表	显示可兑换的商品列表
附近回收	回收机列表	显示当前城市所有回收机及其在地图上的位置
	回收机详情	显示回收机详细信息
	回收机回收	显示回收机业务流程
设置	修改密码	需输入旧密码、新密码，确认密码
	更换手机号	需输入新手机号、验证码或旧密码
公司端	登录	登录、切换用户、退出登录
	处理订单	接单、取消订单、查看订单详情
	个人中心	企业账号的相关信息

拓展实训项目

时代楷模是由中共中央宣传部集中组织宣传的全国重大先进典型。时代楷模充分体现“爱国、敬业、诚信、友善”的社会主义核心价值观，充分体现中华传统美德，是具有很强先进性、代表性、时代性和典型性的先进人物。时代楷模事迹厚重感人、道德情操高尚、影响广泛深远。根据时代楷模的职业身份，以中共中央宣传部和有关部门名义发布时代楷模。在中央电视台设立了“时代楷模发布厅”。

随着经济快速发展，计算机的普及率越来越高，互联网用户数量逐年增多，在多元网络文化环境中，年轻人容易被负面文化影响，误入歧途。时代楷模 App 运用新信息技术，整合多方资源，让更多的年轻人通过该 App 看要闻、学新思想、明历史、长知识、知晓时政，紧跟党的步伐，不断武装思想。它主要包括以下功能模块。

1. 楷模公告：在首页以幻灯片轮播图展示时代楷模精神、往期公告列表。
2. 楷模列表：包括楷模介绍、楷模事迹视频、致敬内容、评论等。
3. 楷模故事：包括楷模事迹介绍、评论等，事迹介绍包括文章、视频等资源。
4. 学习心得：包括学习笔记、学习感言、学习历史等。
5. 公益活动：包括活动发起、活动展示、活动报名等，活动内容包括文章、视频等资源。
6. 身边的楷模：将身边的符合时代楷模定义的事迹发布到平台，传递正能量，事迹内容包括文章、照片、视频等资源。