

# Web作业二手交易平台介绍文档

## 第 18 组

### 一、小组成员与分工

| 成员     | 学号        | 分工   |
|--------|-----------|--|
| 组长：关昕宇 | 191840077 | 1.分派工作，安排进度<br>2.搭建后端基本框架<br>3.重构部分后端代码,修复一些前后端bug |
| 冯亚林    | 191850036 | 1.搭建前端框架，完成了前端的很多工作<br>2.写初始文稿、制作项目录屏              |
| 王涛     | 191850189 | 搭建后端框架，实现了许多功能                                     |
| 夏宇     | 191850205 | 搭建后端框架，实现了许多功能，修复了一些后端bug                          |
| 张林杰    | 191250193 | 搭建前端框架，完成前端的许多工作                                   |

#### 具体分工

##### 前端部分：

**冯亚林：** AppFoot\AppPageBody\AppHeader\Carousel\carouselList\index\platform-admin等

**张林杰：** 其余页面的工作（包括login\login-admin\sign-in\idle-details\ me\ search\ release）等

**关昕宇：** 修复部分前后端交互的bug

前端的其他部分，比如src\api\index.js里封装的向后端发送请求的方法，是由冯亚林和张林杰共同完成的

##### 后端部分：

**关昕宇：** 用户模块和留言模块，收藏功能和搜索功能：登陆注册，查看和编辑个人信息，编辑收货地址；发送留言，消息提示；收藏商品，查看收藏列表；搜索商品名称和介绍。

**夏宇：** 后台管理部分：包括列出所有管理员，添加管理员，列出所有商品，强制修改商品状态，获取所有订单列表，删除订单，获取所有用户列表，更新用户信息功能。后端轮播图管理：包括轮播图的增删改查功能。

**王涛：** 订单模块和商品模块：包括发布商品，购买商品；生成订单，支付订单，订单发货，取消订单。

## 二、项目简介

github地址：<https://github.com/Gxy-2001/NXX>

### 2.1 项目介绍

我们的项目名为NXX(意为“南小咸”，区别于已有的“南小闲”)，是一个Web上的二手交易平台，参考了闲鱼、淘宝和其他的二手交易网站。项目使用了完整的WEB前后端技术，包括SpringBoot、MySQL、Vue、ElementUI等等(见下2.3详细介绍)，完成了二手交易业务的逻辑闭环，包括登录注册、发布闲置、搜索商品、购买商品、评论，还实现了个人中心、网站后台管理等额外的功能。

在开发过程中，因考虑到项目工作量不小，需要学习的地方很多、编码不轻松，所以决定用前后端分离的方式开发，由关昕宇、王涛、夏宇完成后端，由冯亚林、张林杰负责前端。github地址为整合的后端地址，单独的前端地址请见前端Readme。

### 2.2 功能简介

主要功能都在录屏中体现

南小咸项目有以下功能，集中在3个模块上：

- 商品和订单模块：发布商品、购买商品、搜索商品、商品留言、订单生成、订单支付、取消订单等。
- 用户模块：注册、登录、管理个人信息、编辑收货地址、查看个人的各种商品信息（已购商品、收藏、发布清单等）
- 后台管理模块：后台登录、闲置管理、订单管理、用户管理（封号、添加管理员）、首页轮播图管理。

### 2.3 技术简介

前端部分：

前端各个工具的版本在项目文件 Nxx\_web\ package.json 中可以看到

| 核心技术                   | 简要介绍           | 说明  |
|------------------------|----------------|---|
| Vue                    | 核心框架           | 目前最流行的渐进式JavaScript框架。声明式渲染大大便利了开发和测试，组件机制提高了复用性。此外Vue的生态非常丰富，利于开发。<br>版本"vue": "^2.6.11"   |
| Vue-cli                | 项目启动脚手架        | Vue command line interface，基于 Vue.js 进行快速开发的完整系统。<br>版本为@vue/cli 4.5.0  |
| Vue-router             | 路由管理           | Vue 官方的路由管理器。   |
| ElementUI              | 网页UI组件库        | 易用、美观的UI组件库，版本"element-ui": "^2.15.1"   |
| git+github             | 版本控制           | 协同开发必备  |
| Axios实现restful风格       | 前后端交互          | RESTful是目前最流行的 API 设计规范,即向后端发送请求是GET/POST + 宾语的结构<br>Axios 一个基于 promise 的 HTTP 库，可以用在浏览器和 node.js 中。(promise是ES6语法提供的JavaScript对象，用来传递异步信息) |
| Babel, Webpack, jQuery | JavaScript构建工具 | Babel 将TypeScript语法编译为ES语法；<br>Webpack：模块打包器，打包、压缩、合并等，将ES6语法转为ES5，让更多的浏览器能够支持项目<br>jQuery: JavaScript代码库(or JavaScript框架)                  |
| NPM                    |                | 目前主流的NodeJs包管理工具，类似于后端里的maven，通过它来安装和管理各种包  |

## 后端部分

| 主要后端技术              | 说明              | 版本    |
|---------------------|-----------------|-------|
| SpringBoot          | 容器+MVC框架        | 2.4.5 |
| MyBatis             | ORM框架           | 2.1.4 |
| MySQL               | 关系型数据库          | 8     |
| Elasticsearch       | 搜索引擎            | 7.9.3 |
| LogStash-input-jdbc | 数据库和ES间增量同步     | 7.9.3 |
| Swagger-UI          | 后端调试，用Knife4j皮肤 | 2.9.2 |
| Java                | 编程语言            | 1.8   |

## 2.4 项目结构

### 前端

```

├─.idea
│   └─inspectionProfiles
├─dist
│   ├──css
│   ├──fonts
│   ├──img
│   └─js
├─node_modules 属于它的分支全部略去)
├─public
└─src

```

```

├─api
├─assets
│   └─ads
│   └─Background
│   └─Carousel
│       └─style
│           └─purple
│               └─theme
│                   └─fonts
├─components
│   └─common
│   └─page
├─router
└─utils

```

## 后端

|  |               |
|--|---------------|
| ├─doc  | 项目介绍文档和后端开发文档 |
| ├─ES   | ES搜索          |
| ├─sql  | sql文件         |
| ├─src  |               |
| │   └─main   |               |
| │       └─java   |               |
| │           └─com  |               |
| │               └─net  |               |
| │                   └─nxx                                    |               |
| │                       └─common                             |               |
| │                           └─exception                      | 存放异常信息        |
| │                               └─config                     | 配置文件          |
| │                                   └─controller             | Controller层   |
| │                                       └─dao                | 与Mybatis交互接口  |
| │   └─model          | 存放Model       |
| │   └─service    | 提供服务的接口       |
| │   └─impl   | 提供服务的实现       |
| │   └─VO | 数据传递对象        |
| │   └─resources  |               |
| │       └─mapping  | Mybatis实现     |
| │           └─static   | 存放图片          |
| └─test   | 存放测试文件        |
| └─实验报告   |               |

## 2.5 运行步骤

### 后端

1. 安装 java8, Elasticsearch7.9.3, maven等开发环境
2. 执行 `.sql` 目录下的sql文件, 修改配置文件中的数据库密码

3. 开启Elasticsearch，创建名为 `item` 的索引
4. 开启 logstash，安装 `logstash-input-jdbc` 插件，在 `bin` 目录下执行 `\logstash -f ./logstash.conf`，如有问题请见 `.ES` 目录下的readme
5. `import maven`
6. 后端调试页面：[Swagger](#) 或者 [有皮肤的Swagger](#)，可以查看接口

## 前端

为了便于本地部署、运行，推荐安装jetbrains的IDEA（我们用的也是IDEA）。为使用npm，还需要安装nodejs。然后在 `Nxx_Web` 目录下输入如下两条命令：

2. `npm install`
3. `npm run serve`

## 补充

2.1 在idea的terminal输入npm install，以安装各种依赖

因为防火墙，npm下载速度可能不快。这时可以使用npm的淘宝镜像，但最好不要用cnpm，cnpm的安装路径与npm可能不太一样。

可以单次使用淘宝镜像：

```
npm install --registry=https://registry.npm.taobao.org
```

也可以通过下面的命令，将npm的淘宝镜像设置为默认下载源，再npm install

```
$ npm config set registry https://registry.npm.taobao.org
```

·恢复官方默认设置：npm config set registry <https://registry.npmjs.org>

·检测镜像是否安装成功 npm config get registry

2.2 输入npm run serve启动前端

启动成功，会显示

App running at:

- Local: <http://localhost:8080/>

即项目在本地部署的地址，点击打开，即可与系统交互。