

后端技术选型

技术	说明	官网
Spring Boot	容器+MVC框架	https://spring.io/projects/spring-boot
Spring Security	认证和授权框架	https://spring.io/projects/spring-security
MyBatis	ORM框架	http://www.mybatis.org/mybatis-3/zh/index.html
MyBatisGenerator	数据层代码生成	http://www.mybatis.org/generator/index.html
PageHelper	MyBatis物理分页插件	http://git.oschina.net/free/Mybatis_PageHelper
Swagger-UI	文档生产工具	https://github.com/swagger-api/swagger-ui
Elasticsearch	搜索引擎	https://github.com/elastic/elasticsearch
RabbitMq	消息队列	https://www.rabbitmq.com/
Redis	分布式缓存	https://redis.io/
MongoDb	NoSql数据库	https://www.mongodb.com/
zookeeper	分布式锁	https://zookeeper.apache.org/
Docker	应用容器引擎	https://www.docker.com/
Druid	数据库连接池	https://github.com/alibaba/druid
OSS	对象存储	https://github.com/aliyun/aliyun-oss-java-sdk
JWT	JWT登录支持	https://github.com/jwtkt/jjwt
LogStash	日志收集	https://github.com/logstash/logstash-logback-encoder
Lombok	简化对象封装工具	https://github.com/rzwitserloot/lombok
Seata	全局事务管理框架	https://github.com/seata/seata
Portainer	可视化Docker容器管理	https://github.com/portainer/portainer
canal	数据同步	https://github.com/alibaba/canal

前端技术选型

技术	说明	官网
Vue	前端框架	https://vuejs.org/

Vue-router	路由框架	https://router.vuejs.org/
Vuex	全局状态管理框架	https://vuex.vuejs.org/
Element	前端UI框架	https://element.eleme.io/
Axios	前端HTTP框架	https://github.com/axios/axios
v-charts	基于Echarts的图表框架	https://v-charts.js.org/

环境依赖

工具	版本号	下载
JDK	1.8	https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html
Mysql	5.7	https://www.mysql.com/
Redis	3.2	https://redis.io/download
Elasticsearch	6.2.2	https://www.elastic.co/downloads
MongoDb	3.2	https://www.mongodb.com/download-center
RabbitMq	3.7.14	http://www.rabbitmq.com/download.html
nginx	1.10	http://nginx.org/en/download.html

后端开发环境搭建

简易环境搭建流程：

- 安装IDEA并导入项目源码；
- 安装MySQL，创建一个micromall数据库，并导入/document/sql/micromall.sql文件；
- 安装Redis、Elasticsearch、MongoDB、RabbitMQ等环境。

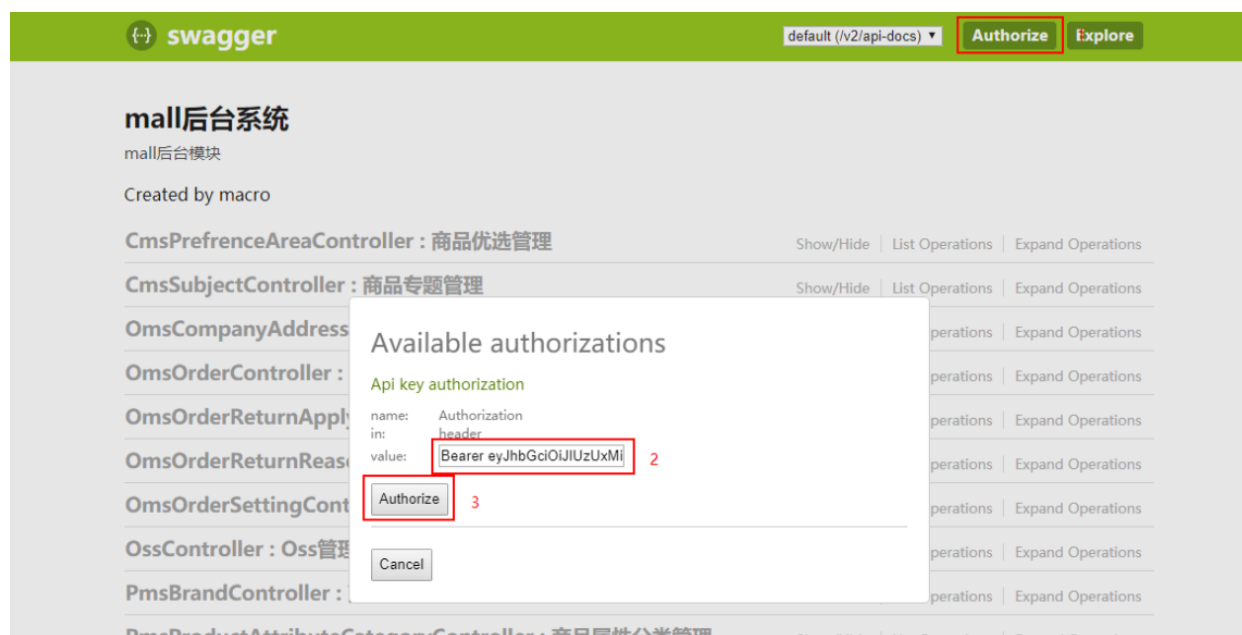
项目部署

到tuling-vip git环境下载后台与前端项目：

- 1、前端项目-用户商城：<http://git.jiagouedu.com/java-vip/tulingmall-front.git>
- 2、前端项目-admin后台管理：<http://git.jiagouedu.com/java-vip/mall-admin-web.git>
- 3、后端项目：<http://git.jiagouedu.com/java-vip/single-mall.git>

启动后台管理服务mall-admin

- 直接运行com.macro.mall.MallAdminApplication的main函数即可；
- 访问接口文档：http://localhost:8081/swagger-ui.html
- 登录接口地址：http://localhost:8081/admin/login，用户admin,密码admin，角色为系统管理员
- 访问登录接口获取到token后放入认证的头信息即可正常访问其他需要登录的接口：



启动前台服务mall-portal

- 直接运行com.macro.mall.portal.MallPortalApplication的main函数即可；
- 访问接口文档：http://localhost:8888/swagger-ui.html
- 登录接口地址：http://localhost:8888/sso/login
- 调用需要登录的接口方式同mall-admin。

这里启动过程中由于依赖rabbitmq，application.yaml文件中的配置

rabbitmq.virtual-host: /mall，启动可能会报错，主要原因我们rabbitmq服务端缺少Virtual Host，需要去创建并且设置对应权限给当前用户

mall前台系统

mall前台模块

Created by macro

HomeController : 首页内容管理	Show/Hide	List Operations	Expand Operations
MemberAttentionController : 会员关注品牌管理	Show/Hide	List Operations	Expand Operations
MemberCollectionController : 会员收藏管理	Show/Hide	List Operations	Expand Operations
MemberReadHistoryController : 会员商品浏览记录管理	Show/Hide	List Operations	Expand Operations
OmsCartItemController : 购物车管理	Show/Hide	List Operations	Expand Operations
OmsPortalOrderController : 订单管理	Show/Hide	List Operations	Expand Operations
OmsPortalOrderReturnApplyController : 申请退货管理	Show/Hide	List Operations	Expand Operations
UmsMemberController : 会员登录注册管理	Show/Hide	List Operations	Expand Operations
UmsMemberCouponController : 用户优惠券管理	Show/Hide	List Operations	Expand Operations
UmsMemberReceiveAddressController : 会员收货地址管理	Show/Hide	List Operations	Expand Operations

[BASE URL: /mall-portal/ , API VERSION: 1.0]

启动搜索服务mall-search

- 直接运行com.macro.mall.search.MallSearchApplication的main函数即可；
- 访问接口文档：http://localhost:8085/swagger-ui.html

mall搜索系统

mall搜索模块

Created by macro

EsProductController : 搜索商品管理		Show/Hide	List Operations	Expand Operations
POST	/esProduct/create/{id}	根据id创建商品		
POST	/esProduct/delete/batch	根据id批量删除商品		
GET	/esProduct/delete/{id}	根据id删除商品		
POST	/esProduct/importAll	导入所有数据库中商品到ES		
GET	/esProduct/recommend/{id}	根据商品id推荐商品		
GET	/esProduct/search	综合搜索、筛选、排序		
GET	/esProduct/search/relate	获取搜索的相关品牌、分类及筛选属性		
GET	/esProduct/search/simple	简单搜索		

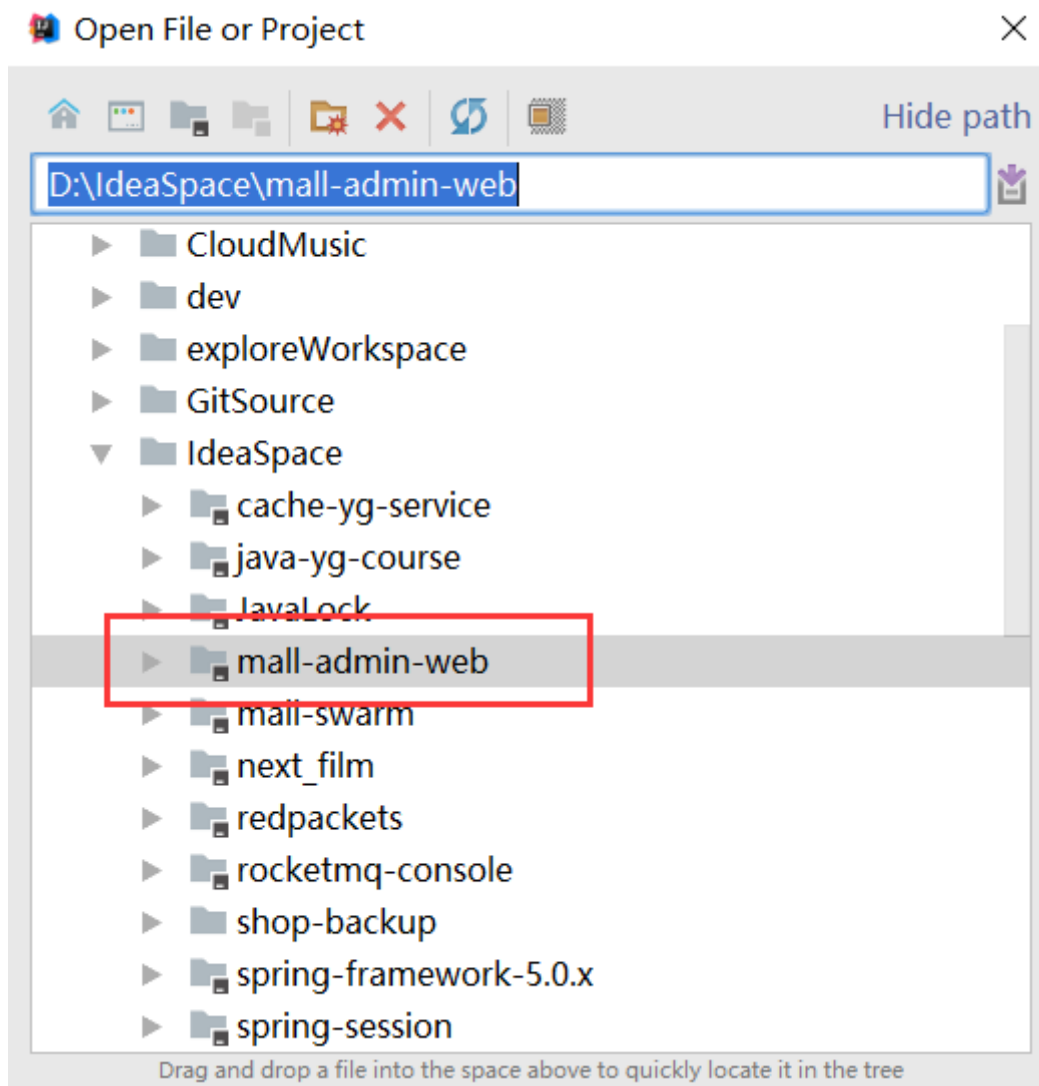
前端项目搭建与部署

前端项目基于Vue+Element实现。需要依赖NodeJS

1、下载安装nodejs

<https://nodejs.org/dist/v8.9.4/node-v8.9.4-x64.msi>, 下载完直接打开安装

2、导入前端项目到IDEA



3、打开控制台安装相关依赖

- 1 打开控制台，执行下面指令
- 2 `D:\IdeaSpace\mall-admin-web>npm install`

执行编译的过程中可能出现，指令不存在的现象。首先检查以下环境变量是否配置正确，如果环境变量正常，一般重启电脑可以解决问题。安装完毕会出现下图红框内的内容

```
Terminal
+ Download complete ] - :
- Binary saved to D:\IdeaSpace\mall-admin-web\node_modules\node-sass\vendor\win32-x64-57\bindir
- Caching binary to C:\Users\39497\AppData\Roaming\npm-cache\node-sass\4.9.0\win32-x64-57_bindir

> uglifyjs-webpack-plugin@0.4.6 postinstall D:\IdeaSpace\mall-admin-web\node_modules\webpack\
> node lib/post_install.js

> node-sass@4.9.0 postinstall D:\IdeaSpace\mall-admin-web\node_modules\node-sass
> node scripts/build.js

Binary found at D:\IdeaSpace\mall-admin-web\node_modules\node-sass\vendor\win32-x64-57\bindir
Testing binary
Binary is fine
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.3 (node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.3: wanted
added 1293 packages in 163.208s
```

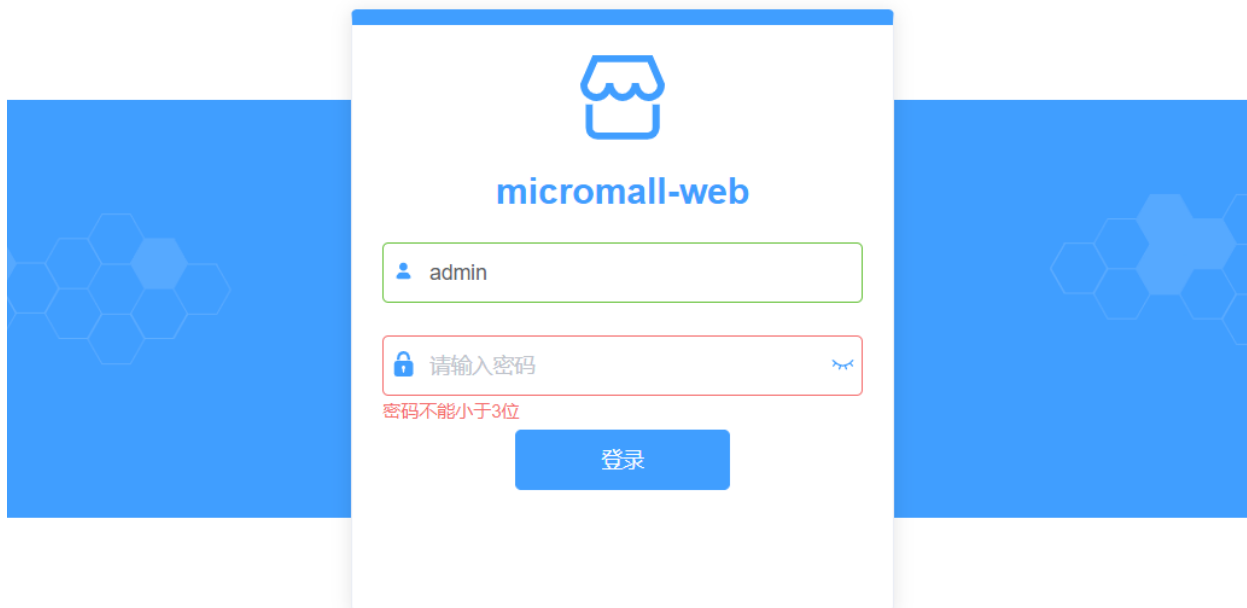
4、启动本地前端服务

- 1 在IDEA控制台中输入如下命令
- 2 `D:\IdeaSpace\mall-admin-web>npm run dev`

启动成功：

```
Terminal
+ DONE Compiled successfully in 15276ms
- I Your application is running here: http://localhost:8090
```

服务访问地址：<http://localhost:8090>



后端服务地址配置在config/dev.env.js当中配置

```
1 module.exports = merge(prodEnv, {  
2   NODE_ENV: '"development"',  
3   //后端地址可在此设置  
4   BASE_API: '"http://localhost:8080"'  
5 })
```

用户商城-前端服务部署

1、启动配置：编辑vue.config.js

```
1 module.exports = {  
2   devServer:{  
3     host:'localhost', //前端服务地址  
4     port:8080, //前端服务端口  
5     proxy:{  
6       '/api':{  
7         // target:'http://localhost:8081',  
8         // target:'http://yangguo.natapp1.cc',  
9         target:'http://localhost:8888', //后端服务uri  
10      changeOrigin:true,  
11      pathRewrite:{  
12        '/api':''  
13      }  
14    }  
15  }
```

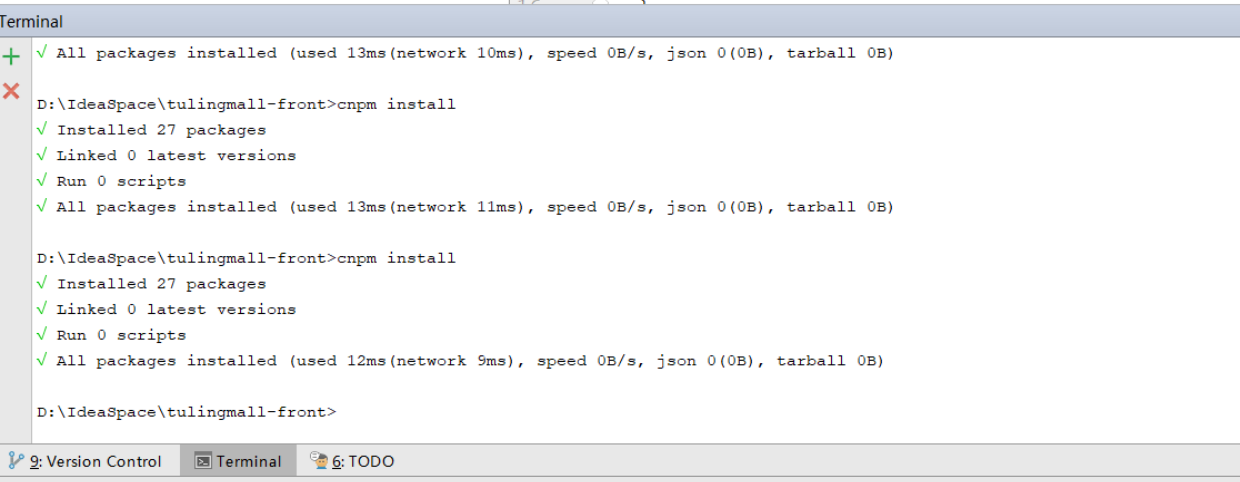
```
14 }  
15 }  
16 },
```

2、编辑完后执行install

- 1、nodejs 默认的命令执行为: `npm install` //下载依赖太慢, 容易出错
- 2 建议安装taobao镜像
- 3 2、安装命令: `npm install -g cnpm --registry=https://registry.npm.taobao.org`
- 4 安装完毕执行install, 成功的结果如下图所示
- 5 3、`cnpm install` //cnpm需要安装插件
- 6 4、`cnpm run serve` //启动前端服务器

install结果:

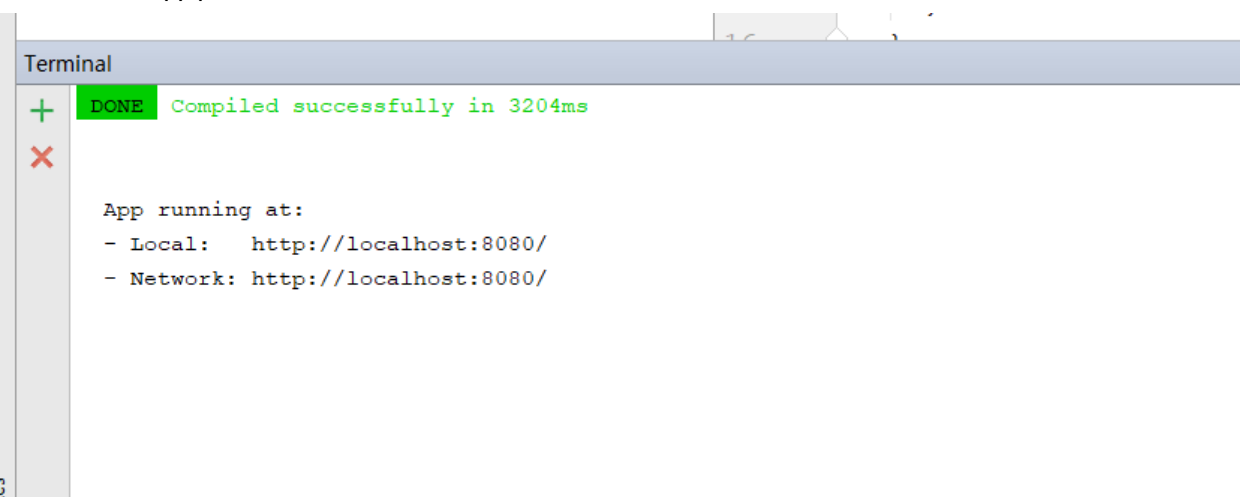
```
14 }  
15 }  
16 },
```



The terminal window shows the following output:

```
+ All packages installed (used 13ms(network 10ms), speed 0B/s, json 0(0B), tarball 0B)  
D:\IdeaSpace\tulingmall-front>cnpm install  
✓ Installed 27 packages  
✓ Linked 0 latest versions  
✓ Run 0 scripts  
✓ All packages installed (used 13ms(network 11ms), speed 0B/s, json 0(0B), tarball 0B)  
  
D:\IdeaSpace\tulingmall-front>cnpm install  
✓ Installed 27 packages  
✓ Linked 0 latest versions  
✓ Run 0 scripts  
✓ All packages installed (used 12ms(network 9ms), speed 0B/s, json 0(0B), tarball 0B)  
  
D:\IdeaSpace\tulingmall-front>
```

run serve 结果:



The terminal window shows the following output:

```
+ DONE Compiled successfully in 3204ms  
  
App running at:  
- Local: http://localhost:8080/  
- Network: http://localhost:8080/
```

有道云笔记链接:

<http://note.youdao.com/noteshare?id=c93192f6085f87f2c1c9fc40c5235bf9&sub=BDAE83744E474E8E87D69CE63B3F4BC6>