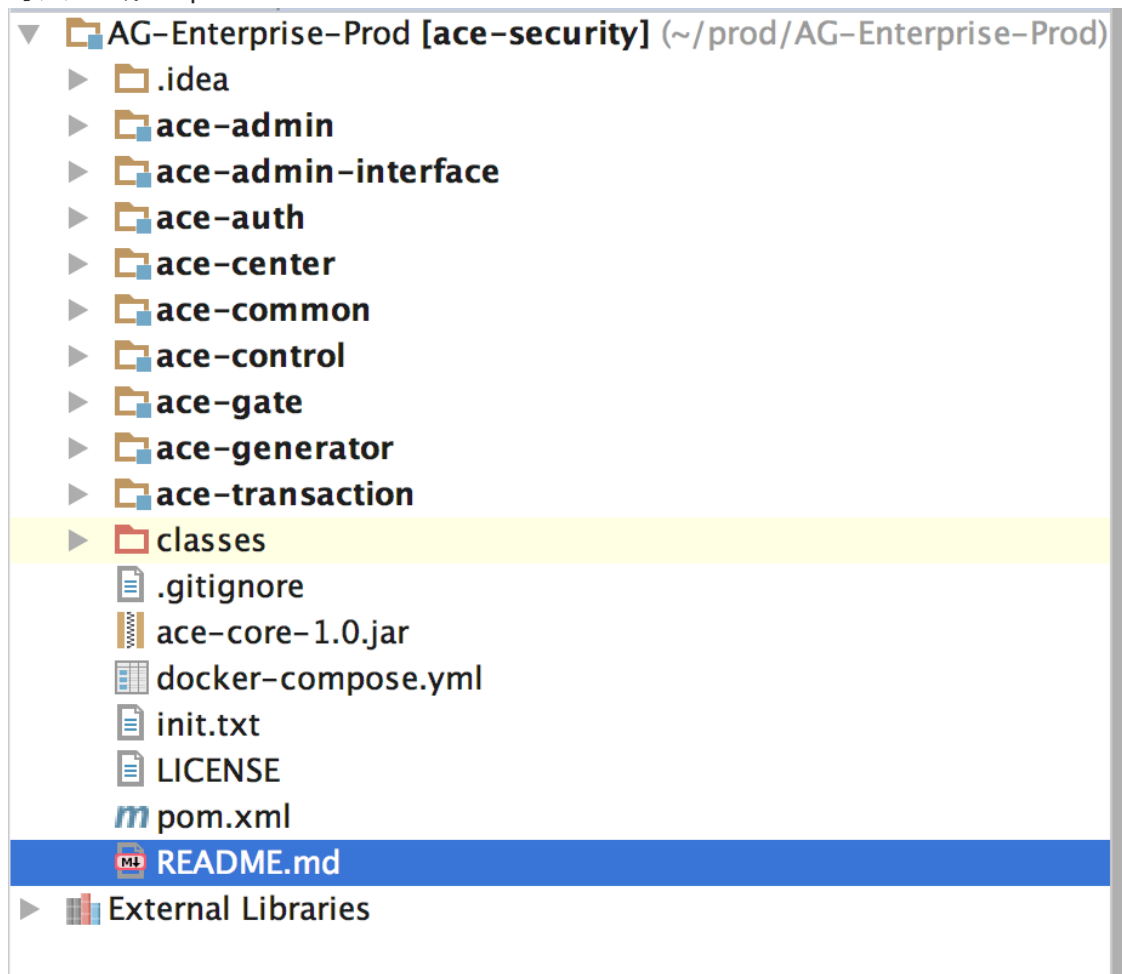


后端工程启动

下载AG-Enterprise-Prod代码

导入ide或eclipse



项目结构

```
├── ace-security
│   │
│   └── ace-module-----业务服务模块
│       ├── -ace-admin:8762-----基础管理后台服务
│       ├── -ace-dict:9999-----基础数据字典服务
│       └── -ace-workflow:8082-----基础 workflow 服务
```

			-ace-tool:9991-----	搜索、oss存储
			-ace-interface-----	基础服务数据对象
			-ace-auth-----	鉴权模块
			-ace-auth-client-----	鉴权客户端SDK
			-ace-auth-server:9777-----	鉴权中心服务
			-ace-gate-----	网关负载中心
			-ace-gate-ratelimit-----	网关限流插件
			-ace-gate-server:8765-----	网关中心服务
			-ace-center:8761-----	服务注册中心
			-ace-config:8750-----	统一配置中心
			-ace-control-----	统一监控中心
			-ace-monitor:8764-----	服务监控中心
			-ace-trace:9411-----	链路追踪中心
			-ace-base-----	基础工具模块
			-ace-common-----	boot脚手架工具
			-ace-core-----	核心模块
			-ace-merge-parent-----	数据聚合工具
			-ace-generator-----	通用代码生成器
			-ace-transaction-----	分布式事务中心
			-ace-job-----	定时作业中心
			-ace-demo-----	示例模块

环境须知

- maven，建议配置阿里云镜像
- mysql一个，redis一个，rabbitmq一个
- jdk1.8
- IDE插件一个， `lombok`插件，具体百度即可

运行步骤

- 编译和装载工程 在工程根目录执行如下命令

```
mvn clean install -Dmaven.test.skip=true
```

- 运行数据库脚本：DB/下所有脚本
- 修改配置：
 1. ace-admin/src/main/resources/application.yml（数据库、rabbit mq、redis）
 2. ace-gate-server/src/main/resources/application.yml（rabbit mq、redis）
 3. ace-auth-server/src/main/resources/application.yml（数据库、rabbit mq、redis）
 4. ace-dict/src/main/resources/application.yml（数据库、rabbit mq、redis）
- 按 顺序 运行核心模块：CenterBootstrap（ace-center）、ConfigServerBootstrap（ace-config）、AuthBootstrap（ace-auth-server）、DictBootstrap（ace-dict）、AdminBootstrap（ace-admin）、GateBootstrap（ace-gate）
- 根据需要启动非核心模块：TxManagerApplication（tx-manager）、TraceBootstrap（ace-control/ace-trace）、MonitorBootstrap（ace-monitor）

前端工程启动（确认后端正确启动）

下载AG-Enterprise-UI代码

推荐使用开发工具 `vscode`

环境搭建

```
node 版本: v6.11.2  
npm 版本: 3.10.10
```

开发

在前端工程的根目录下运行下面命令

```
# 安装依赖
npm install
//or # 建议不要用cnpm 安装有各种诡异的bug 可以通过如下操作解决npm速度慢的问题
npm install --registry=https://registry.npm.taobao.org

# 本地开发 开启服务
npm run dev
```

浏览器访问: <http://localhost:9527>

发布

在前端工程的根目录下运行下面命令, 通过下面命令, 会在更目录下产生一个 `dist` 的文件夹, 里面就是我们编译后的生产代码, 可以直接用nginx部署。

```
# 发布测试环境 带webpack analyzer
npm run build:sit-preview

# 构建生成环境
npm run build:prod
```

目录结构

— build	// 构建相关
— config	// 配置相关
— src	// 源代码
— api	// 所有请求
— assets	// 主题 字体等静态资源
— components	// 全局公用组件
— directive	// 全局指令
— filtres	// 全局filter
— mock	// mock数据
— router	// 路由

```
|   |—— store           // 全局store管理
|   |—— styles         // 全局样式
|   |—— utils          // 全局公用方法
|   |—— view           // view
|   |—— App.vue        // 入口页面
|   |—— main.js        // 入口 加载组件 初始化等
|—— static             // 第三方不打包资源
|   |—— Tinymce        // 富文本
|—— .babelrc           // babel-loader 配置
|—— eslintrc.js        // eslint 配置项
|—— .gitignore         // git 忽略项
|—— favicon.ico        // favicon图标
|—— index.html         // html模板
|—— package.json       // package.json
```