# 安装指南

本章节介绍如何本地安装运行Mango权限管理系统，本书着眼于实践，建议读者结合源码和书本内容逐步学习和掌握Mango系统的整个开发过程。因为是前后端分离项目，所以，项目安装指南分为前端安装指南和后端安装指南两部分。

## 2.1 前端安装指南

本系统前端使用Vue.js和Element框架搭建，基于NPM环境开发，所以在开发之前，需要先安装Node.JS，开发工具为Visual Studio Code，当然读者也可以根据自己的喜好选择其他开发工具，比如说WebStorm，具体开发环境的搭建请参考后续章节：前端篇◎搭建开发环境，这里着重说明以下现有源码的安装和运行。

### 2.1.1 开发环境

前端开发环境基于NPM环境，使用VS Code开发。

IDE : VS Code 1.27

NODE: Node 10.15.x

NPM : NPM 6.4.x

…

### 2.1.2 技术选型

前端技术主要使用Vue.js和Elment UI框架。

前端框架：Vue 2.x

页面组件：Element 2.x

状态管理：Vuex 2.x

后台交互：axios 0.18.x

图标使用：Font Awesome 4.x

…

### 2.1.3 项目结构

前端项目结构如下。

mango-ui

-- build：项目编译相关模块，项目模板自动生成

-- config：项目配置相关模块，项目模板自动生成

-- src：项目源码模块，前端开发工作集中在此目录

-- assets： 图标、字体、国际化信息等静态信息

-- components： 组件库，对常用组件进行封装

-- http： 后台交互模块，统一后台接口请求API

-- i18n： 国际化模块，使用Vue i18n进行国际化

-- mock： Mock模块，模拟接口调用并返回定制数据

-- permission： 权限控制模块，处理权限认证逻辑

-- router： 路由管理模块，负责页面各种路由配置

-- store： 状态管理模块，提供组件间状态共享

-- utils： 工具模块，提供一些通用的工具方法

-- views： 页面模块，主要放置各种页面视图组件

### 2.1.4 编译运行

#### 获取源码

获取前端源码，整个前端只有一个工程mango-ui，将其拷贝放置到本地目录。

#### 编译源码

在mango-ui目录下打开CMD终端，执行 npm install, 下载和安装项目依赖包。

#### 启动系统

执行 npm run dev 命令，启动项目，启动之后通过 http://localhost:8080 访问。

#### 项目打包

执行 npm run build 命令，进行前端项目打包，打包完成之后会生成 dist 目录。

将生成的目录直接放置到如Tomcat之类的WEB服务器，启动服务即可访问。

#### Mock 开关

本系统采用前后端分离架构，前端若开启Mock模块，可以模拟大部分接口数据。

通过修改src/mock/index.js中的openMock变量，可以一键开启或关闭Mock功能。

#### 修改配置

如果想自定义端口（默认是8080），可以修改 config/index.js 下的 port 属性。

后台接口和备份服务器地址配置在 src/utils/global.js，如有修改请做相应变更。

## 2.2后端安装指南

### 2.2.1开发环境

后端开发环境基于Java环境，使用Eclipse开发。

IDE : eclipse 4.x

JDK : JDK1.8.x

Maven : Maven 3.5.x

MySQL: MySQL 5.7.x

Consul: Consul 1.4.0

…

### 2.2.2技术选型

后端技术主要使用Spring Boot、Spring Cloud和MyBatis框架。

核心框架：Spring Boot 2.x

服务治理：Spring Cloud Finchley

安全框架：Spring Security 5.x

视图框架：Spring MVC 5.x

持久层框架：MyBatis 3.x

数据库连接池：Druid 1.x

消息队里：RabbitMQ

接口文档：Swagger 2.9.x

日志管理：SLF4J、Log4j

…

### 2.2.3项目结构

后端项目源码工程结构。

mango-common： 公共代码模块，主要放置一些工具类

mango-core： 封装业务模块，主要封装公共业务模块

mango-admin： 后台管理模块，包含用户、角色、菜单管理等

mango-backup： 系统数据备份还原模块，可选择独立部署

mango-monitor： 系统监控服务端，监控Spring Boot应用

mango-producer： 服务提供者示例，方便在此基础上搭建模块

mango-consumer： 服务消费者示例，方便在此基础上搭建模块

mango-hystrix： 服务熔断监控模块，收集汇总熔断统计信息

mango-zuul： API服务网关模块，统一管理和转发外部调用请求

mango-config： 配置中心服务端，生成GIT配置文件的访问接口

mango-consul： 注册中心，安装说明目录，内附安装引导说明

mango-zipkin： 链路追踪，安装说明目录，内附安装引导说明

config-repo： 配置中心仓库，在GIT上统一存储系统配置文件

mango-pom： 聚合模块，仅为简化打包，一键执行打包所有模块

### 2.2.4编译运行

#### 获取源码

获取后端源码，获取上面所列所有项目结构，将其拷贝放置到本地目录。

#### 导入工程

使用 Eclipse导入 Maven 项目，在此之前请确认已安装 JDK 和 Maven 工具。

#### 编译源码

找到 mango-pom 工程下的 pom.xml，执行 maven clean install 命令进行一键打包。

一般来说不会有什么问题，如果还打包失败，可以按照优先级逐个编译试一试。

#### 导入数据库

新建mango数据库，使用项目sql目录下的mango.sql 脚本，导入初始化数据库。

修改 mango-admin 下 application.yml 中的数据源配置信息为自己的数据库配置。

修改 mango-backup下 application.yml 中的数据源配置信息为自己的数据库配置。

#### 启动系统

##### 基础必需模块（注册中心：mango-consul，服务监控：mango-monitor）

找到 mango-consul 工程，根据安装说明安装注册中心，执行 consul agent -dev 启动。

找到 mango-monitor 工程下的MangoMonitorApplication， 启动项目，开启服务监控。

##### 权限管理模块（权限管理：mango-admin，备份还原：mango-backup）

找到 mango-admin 工程下的MangoAdminApplication， 启动项目，开启权限系统服务。

找到 mango-backup 工程下的MangoBackupApplication.java，启动项目，开启备份服务。

##### 其他示例模块（Spring Cloud示例模块，作为开发模板和范例，根据需要启动）

以下为Spring Cloud体系各种功能的实现范例，可以根据需要启动，后续扩展开发也可以作为参考和模板使用，具体使用教程请参考本书后面Spring Cloud系列教程的章节，关于Spring Cloud体系的各种功能模块都有详细的讲解和完整的案例实现。

这些示例模块包括：

mango-producer： 服务提供者示例，演示服务提供者的实现

mango-consumer： 服务消费者示例，演示服务消费者的实现

mango-hystrix： 服务熔断监控模块，演示熔断监控功能的实现

mango-zuul： API服务网关模块，演示API统一网关的实现

mango-config： 配置中心服务端，演示分布式配置中心的实现

注意事项：

注册中心是基础服务，需要先安装Consul，找到mango-consul工程，根据安装说明安装Consul。

如果需要链路追踪服务，需要安装zipkin，找到mango-zipkin 工程，根据安装说明安装zipkin。

如果需要配置中心服务，需要安装rabbitMQ，找到mango-config 工程，根据安装说明安装rabbitMQ。