

环境部署

准备工作

- 1
- JDK >= 1.8 (推荐1.8版本)
- 2
- Mysql >= 5.7.0 (推荐5.7版本)
- 3
- Redis >= 3.0
- 4
- Maven >= 3.0
- 5
- Node >= 12

提示

前端安装完node后，最好设置下淘宝的镜像源，不建议使用cnpm（可能会出现奇怪的问题）

运行系统

前往 Gitee 下载页面(https://gitee.com/y_project/RuoYi-Vue)下载解压到工作目录

后端运行

- 1、导入到 Eclipse ， 菜单 File -> Import ， 然后选择 Maven -> Existing Maven Projects ， 点击 Next > 按钮，选择工作目录，然后点击 Finish 按钮，即可成功导入。
Eclipse 会自动加载 Maven 依赖包，初次加载会比较慢（根据自身网络情况而定）
- 2、创建数据库 ry-vue 并导入数据脚本 ry_2021xxxx.sql ， quartz.sql
- 3、打开项目运行 com.ruoyi.RuoYiApplication.java ， 出现如下图表示启动成功。

1

(♥◡‿◡)ﾉ* 若依启动成功 *ㄟ(ˊᆞˋ)ㄟ

2

.....

3

| _ _ \ \ \ / /

4

| (') | \ _ . / '

5

| (_ o _) / _ () _ . '

6

| (_ , _) . ' _ _ (_ o _) '

7

| | \ \ | | | (_ , _) '



提示

后端运行成功可以通过(<http://localhost:8080>)访问,但是不会出现静态页面,可以继续参考下面步骤部署 `ruoyi-ui` 前端,然后通过前端地址来访问。

前端运行

```
1 # 进入项目目录
2 cd ruoyi-ui
3
4 # 安装依赖
5 npm install
6
7 # 强烈建议不要用直接使用 cnpm 安装,会有各种诡异的 bug,可以通过重新指定 registry
8 npm install --registry=https://registry.npmmirror.com
9
10 # 本地开发 启动项目
11 npm run dev
```

sh

4、打开浏览器,输入: (<http://localhost:80>) 默认账户/密码 `admin/admin123`)

若能正确展示登录页面,并能成功登录,菜单及页面展示正常,则表明环境搭建成功

建议使用 `Git` 克隆,因为克隆的方式可以和 `RuoYi` 随时保持更新同步。使用 `Git` 命令克隆

```
1 git clone https://gitee.com/y_project/RuoYi-Vue.git
```

提示

因为本项目是前后端完全分离的,所以需要前后端都单独启动好,才能进行访问。
前端安装完node后,最好设置下淘宝的镜像源,不建议使用cnpm(可能会出现奇怪的问题)



必要配置

- 修改数据库连接，编辑 `resources` 目录下的 `application-druid.yml`

```
1  # 数据源配置
2  spring:
3      datasource:
4          type: com.alibaba.druid.pool.DruidDataSource
5          driverClassName: com.mysql.cj.jdbc.Driver
6          druid:
7              # 主库数据源
8              master:
9                  url: 数据库地址
10                 username: 数据库账号
11                 password: 数据库密码
```

- 修改服务器配置， 编辑`resources`目录下的`application.yml`

```
1  # 开发环境配置
2  server:
3      # 服务器的HTTP端口，默认为80
4      port: 端口
5      servlet:
6          # 应用的访问路径
7          context-path: /应用路径
```

部署系统

提示

因为本项目是前后端完全分离的，所以需要前后端都单独部署好，才能进行访问。

后端部署

- 打包工程文件



提示

多模块版本会生成在 `ruoyi/ruoyi-admin` 模块下 `target` 文件夹

• 部署工程文件

1、jar部署方式

使用命令行执行: `java -jar ruoyi.jar` 或者执行脚本: `ruoyi/bin/run.bat`

2、war部署方式

`ruoyi/pom.xml` 中的 `packaging` 修改为 `war` , 放入 `tomcat` 服务器 `webapps`

```
1      <packaging>war</packaging>
```

提示

多模块版本在 `ruoyi/ruoyi-admin` 模块下修改 `pom.xml`

- SpringBoot 去除内嵌 Tomcat (PS: 此步骤不重要, 因为不排除也能在容器中部署 war)

```
1      <!-- 多模块排除内置tomcat -->
2      <dependency>
3          <groupId>org.springframework.boot</groupId>
4          <artifactId>spring-boot-starter-web</artifactId>
5          <exclusions>
6              <exclusion>
7                  <groupId>org.springframework.boot</groupId>
8                  <artifactId>spring-boot-starter-tomcat</artifactId>
9              </exclusion>
10         </exclusions>
11     </dependency>
12
13     <!-- 单应用排除内置tomcat -->
14     <exclusions>
15         <exclusion>
16             <artifactId>spring-boot-starter-tomcat</artifactId>
17             <groupId>org.springframework.boot</groupId>
```



前端部署

当项目开发完毕，只需要运行一行命令就可以打包你的应用

```
1 # 打包正式环境
2 npm run build:prod
3
4 # 打包预发布环境
5 npm run build:stage
```

sh

构建打包成功之后，会在根目录生成 `dist` 文件夹，里面就是构建打包好的文件，通常是 `***.js` 、 `***.css` 、 `index.html` 等静态文件。

通常情况下 `dist` 文件夹的静态文件发布到你的 `nginx` 或者静态服务器即可，其中的 `index.html` 是后台服务的入口页面。

outputDir 提示

如果需要自定义构建，比如指定 `dist` 目录等，则需要通过 `config` 的 `outputDir` 进行配置。

publicPath 提示

部署时改变页面js 和 css 静态引入路径,只需修改 `vue.config.js` 文件资源路径即可。

```
1 publicPath: './' //请根据自己路径来配置更改
2
3
1 export default new Router({
2   mode: 'hash', // hash模式
3 })
```

js

js

环境变量



它们都会通过 `webpack.DefinePlugin` 插件注入到主包。

环境变量必须以 `VUE_APP_` 为开头。如: `VUE_APP_API` 、 `VUE_APP_TITLE`

你在代码中可以通过如下方式获取:

```
1 console.log(process.env.VUE_APP_xxxx)
```

js

扩展阅读: 《[Vue CLI - 环境变量和模式](#)》

注意

环境配置修改后, 需要重新运行才会生效

Tomcat配置

修改 `server.xml` , `Host` 节点下添加

```
1 <Context docBase="" path="/" reloadable="true" source=""/>
```

`dist` 目录的文件夹下新建 `WEB-INF` 文件夹, 并在里面添加 `web.xml` 文件

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
3         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4         xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
5         http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
6         version="3.1" metadata-complete="true">
7     <display-name>Router for Tomcat</display-name>
8     <error-page>
9         <error-code>404</error-code>
10        <location>/index.html</location>
11    </error-page>
12 </web-app>
```

Nginx配置

```
1 worker_processes 1;
2
3 events {
4     worker_connections 1024;
5 }
6
7 http {
8     include mime.types;
9     default_type application/octet-stream;
10    sendfile on;
11    keepalive_timeout 65;
12
13    server {
14        listen 80;
15        server_name localhost;
16        charset utf-8;
17
18        location / {
19            root /home/ruoyi/projects/ruoyi-ui;
20            try_files $uri $uri/ /index.html;
21            index index.html index.htm;
22        }
23
24        location /prod-api/ {
25            proxy_set_header Host $http_host;
26            proxy_set_header X-Real-IP $remote_addr;
27            proxy_set_header REMOTE-HOST $remote_addr;
28            proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
29            proxy_pass http://localhost:8080/;
30        }
31
32        error_page 500 502 503 504 /50x.html;
33        location = /50x.html {
34            root html;
35        }
36    }
37 }
```



常见问题

1. 如果使用 Mac 需要修改 `application.yml` 文件路径 `profile`
2. 如果使用 Linux 提示表不存在, 设置大小写敏感配置在 `/etc/my.cnf` 添加
`lower_case_table_names=1` , 重启MYSQL服务
3. 如果提示当前权限不足, 无法写入文件请检查 `application.yml` 中的 `profile` 路径或
`logback.xml` 中的 `log.path` 路径是否有可读可写操作权限

如遇到无法解决的问题请到[Issues](#) 反馈, 会不定时进行解答。

[← 快速了解](#)

[项目介绍 →](#)