



# WAR部署方案

## 正式环境部署

- 部署方案采用nginx+tomcat部署方案
- 后端服务发布部署到tomcat中
- 前端项目由于build后都是静态问题，部署到nginx中

### 一、后台项目jeecg-boot打war包（jeecg-boot-module-system）

（1）后台项目jeecg-boot-module-system打war包之前要进行如下改动

1、pom.xml文件中项目打包格式设置为war

war

具体配置如下：

```
<modelVersion>4.0.0</modelVersion>
<groupId>org.jeecgframework.boot</groupId>
<artifactId>jeecg-boot-module-system</artifactId>
<version>2.0.0</version>
<packaging>war</packaging>
```

2、pom.xml文件删除插件spring-boot-maven-plugin

下面配置删除

```
<build>
  <plugins>
    <plugin>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-maven-plugin</artifactId>
    </plugin>
  </plugins>
</build>
```

3、增加项目web容器部署的支持：

修改类/src/main/java/org/jeecg/JeecgApplication.java

代码如下：

```
package org.jeecg;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.boot.builder.SpringApplicationBuilder;
import org.springframework.boot.web.servlet.support.SpringBootServletInitializer;

import springfox.documentation.swagger2.annotations.EnableSwagger2;

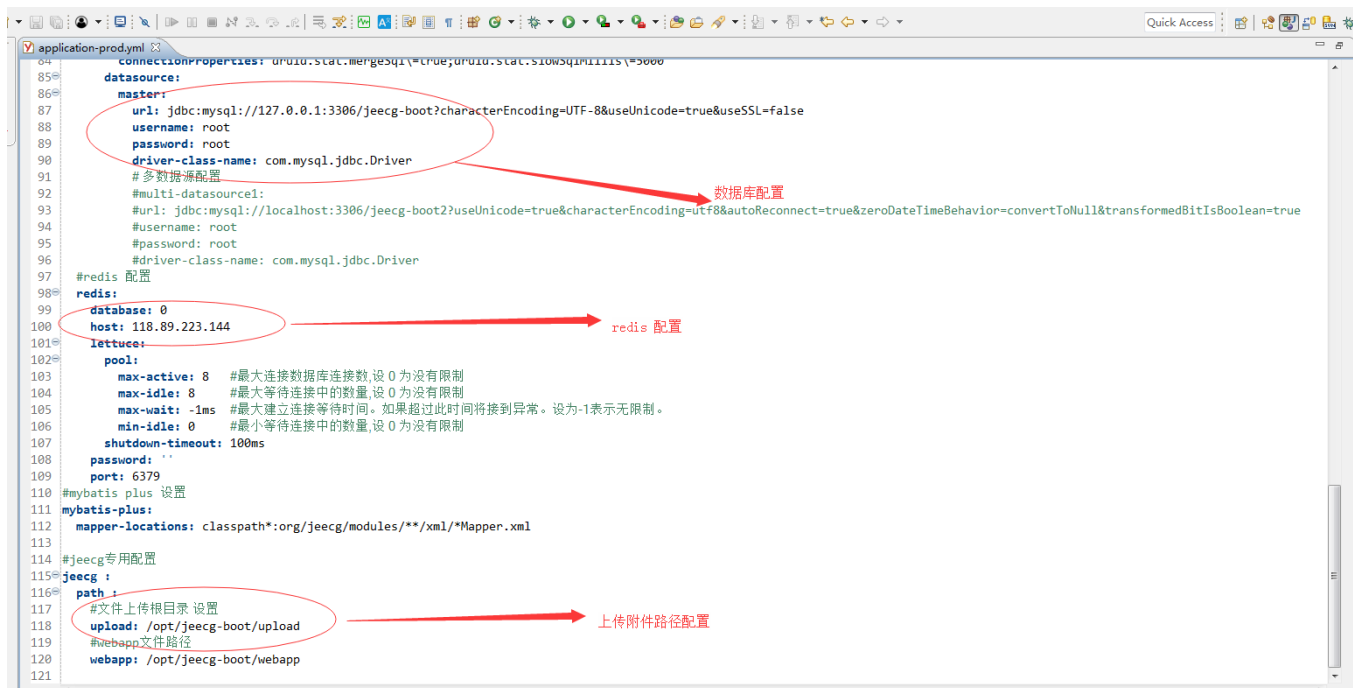
@SpringBootApplication
@EnableSwagger2
public class JeecgApplication extends SpringBootServletInitializer {

    @Override
    protected SpringApplicationBuilder configure(SpringApplicationBuilder application) {
        return application.sources(JeecgApplication.class);
    }

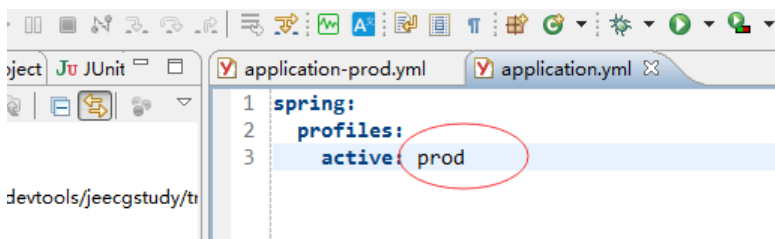
    public static void main(String[] args) {
        System.setProperty("spring.devtools.restart.enabled", "true");
        SpringApplication.run(JeecgApplication.class, args);
    }
}
```

#### 4、修改配置文件（数据库和redis配置）

- 1、修改数据库连接 application-prod.yml
- 2、修改缓存redis配置 application-prod.yml
- 3、修改上传附件配置 application-prod.yml



- 4、切换配置为线上配置 application.yml



然后 maven package 打war包

## 二、后台项目jeecg-boot部署tomcat

- 1、设置tomcat端口号 8080，设置tomcat编码 URIEncoding="UTF-8"
  - 2、部署项目到tomcat安装目录webapps/jeecg-boot工程目录下
- 部署完后通过http://localhost:8080/jeecg-boot 可以访问项目，提示token错误说明部署成功！！

## 三、前台项目build

- 1、修改 public/index.html

```
//图片预览请求地址
window._CONFIG['domianURL'] = 'http://localhost:8080/jeecg-boot';
window._CONFIG['imgDomainURL'] = 'http://localhost:8080/jeecg-boot/sys/common/view';
```

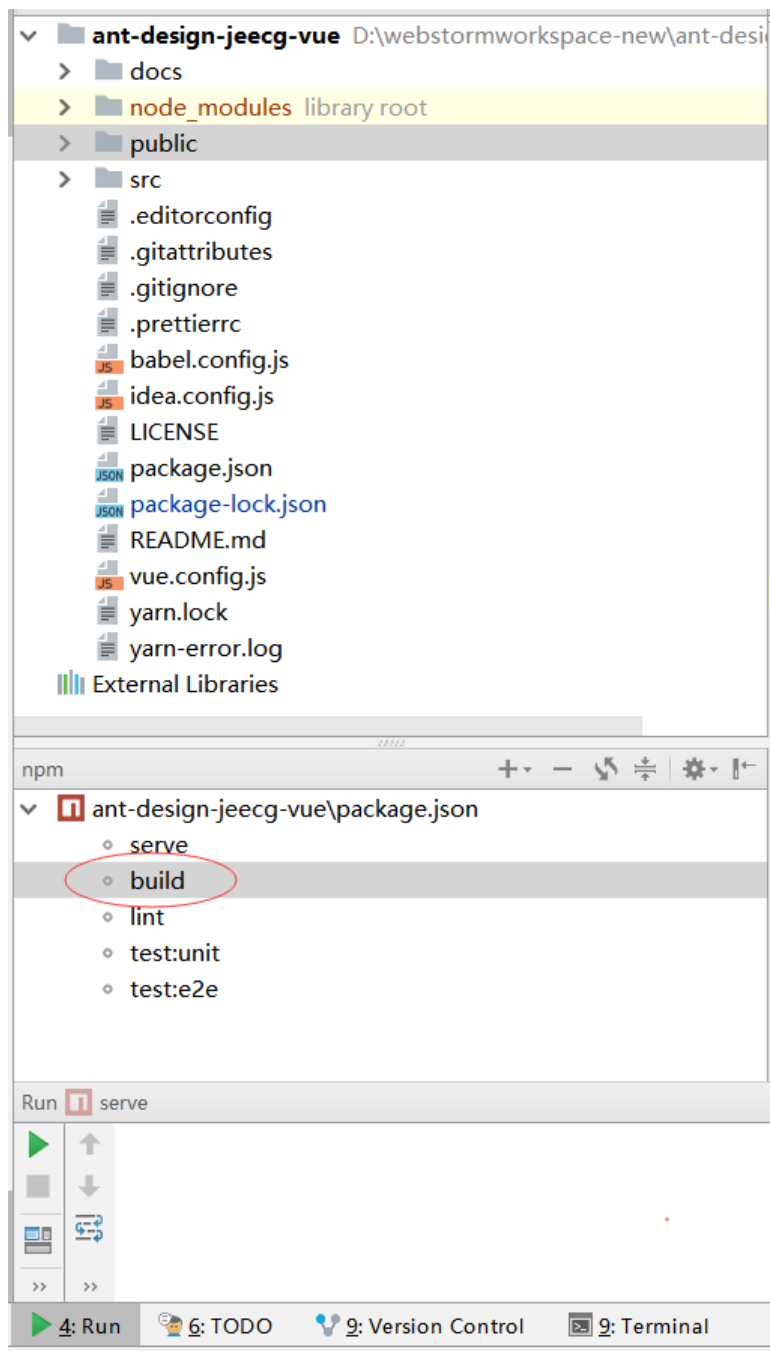
- 2、后台接口服务项目名默认是jeecg-boot，如果需要个性化可以修改src/utlis/request.js 中baseUrl参数（一般情况下默认不需要修改）

具体代码如下：

```
// 创建 axios 实例
const service = axios.create({
  baseUrl: '/jeecg-boot/', // api base_url
  timeout: 6000 // 请求超时时间
})
```

### 3、build项目

使用build命令打包项目



build完成后会生成一个dist的目录该目录下即为build后的文件。

### 4、nginx部署前端项目

拷贝dist下的代码到nginx安装目录下html目录中，即可

### 四、nginx配置（conf/nginx.conf）

nginx监听80端口

```
server {  
    listen      80;  
    server_name 你的域名;  
  
    #后台服务配置，配置了这个location便可以通过http://域名/jeecg-boot/xxxx 访问  
    location ^~ /jeecg-boot {  
        proxy_pass          http://127.0.0.1:8080/jeecg-boot/;  
        proxy_set_header    Host 127.0.0.1;  
        proxy_set_header    X-Real-IP $remote_addr;  
        proxy_set_header    X-Forwarded-For $proxy_add_x_forwarded_for;  
    }  
    #解决Router(mode: 'history')模式下，刷新路由地址不能找到页面的问题  
    location / {  
        root    html;  
    }  
}
```


```
index index.html index.htm;  
if (!-e $request_filename) {  
    rewrite ^(.*)$ /index.html?s=$1 last;  
    break;  
}  
}  
}
```

## 五、nginx 开启压缩，提高首页访问效率

<https://github.com/zhangdaiscott/jeecg-boot/issues/88>

配置后启动tomcat，启动nginx

通过<http://你的域名/> 访问项目，出现如下页面，使用账户/密码：admin/123456 登录成功即可





# Ant Design

Ant Design 是西湖区最具影响力的 Web 设计规范

账号密码登陆

手机号登陆

 admin




 .....

☒ 自动登陆

忘记密码

确 定

其他登陆方式



注册账户

帮助

隐私

条款

Copyright © 2018 JEECG开源社区出品

上一篇：[JAR部署方案](#)

下一篇：[极简部署方案](#)