

# OnlineJudge 前端二次开发指北

Hatsune Miku

2018 年 5 月 12 日

## 1 简介

OnlineJudge<sup>1</sup>是由青岛大学开发的一套在线程序评测系统。具有界面美观、运行可靠、使用了大量现代化的框架与技术、功能强大等特点。但是对其进行二次开发比较困难。单单是前端部分就需要比较复杂的操作。而对此官方又没有一份比较详细的文档，更是加剧了开发的困难。这篇文档简单介绍了 OnlineJudge 在线程序评测系统前端部分的二次开发部署流程。希望对大家的二次开发有所帮助<sup>2</sup>。

整篇文档假设读者已经获得了 OnlineJudge 的源代码，并且了解 Linux 的基本操作。

## 2 开始

在修改完成前端代码后，您一定非常希望运行一下代码看看效果。首先建议按照 Readme 文件建立起一个开发服务器进行一下调试，流程如下：

1. 打开一个终端窗口，并切换到项目目录下；
2. 运行 `npm install`，安装项目依赖，可能需要较长时间；
3. 运行 `NODE_ENV=development npm run build:dll` 构建 webpack 的相关文件，注意**如果没有大的更新就不需要每次都运行了，只需要运行一次**；
4. 运行 `export TARGET=http://Your-backend`，注意这个地方**不需要进行任何修改，不要被这个路径迷惑了！**；
5. 运行 `npm run dev`，建立一个前端服务器。

然后就能在 `localhost:8080` 看到前端的效果图了，由于还没有连接到后端，所以可能几乎什么也不能做。但至少能看看能不能用，如果是翻译的话，可以检查有没有编码问题之类的。

测试之后，就可以用 `Ctrl + C` 关闭测试服务器了。

## 3 投入生产环境

在测试没有什么大问题之后，我们就可以把前端界面投入生产环境了。流程如下：

---

<sup>1</sup>项目地址：<https://github.com/QingdaoU/OnlineJudge>

<sup>2</sup>作者：某弱校 OIer，GitHub 主页：<https://github.com/MingqiHuang>

1. 运行`NODE_ENV=production npm run build:dll`构建 webpack 的相关文件, 开发与生产环境是不一样的, 要注意**如果没有大的更新就不需要每次都运行了, 只需要运行一次**;
2. 运行`npm run build`, 构建项目, 可能需要较长时间;
3. 运行`docker cp ./dist oj-backend:/app/`, 将新编译好的代码注入到前端里。这时如果您刷新您的 Online Judge, 您可能会发现 Online Judge 一直在加载, 加载到天昏地暗。不用担心, 我们有解决方案:

1. 运行`docker exec -it oj-backend /bin/sh`, 切换到 Online Judge 的终端提示符下;
2. 在形如`# /app` 的提示符下, 运行`cd deploy/`, 切换一下目录;
3. 运行`./entrypoint.sh`, 重新启动服务器, 您会得到类似以下的输出:

```
Django VERSION (1, 11, 4, 'final', 0)
Operations to perform:
Apply all migrations: account, announcement, auth, conf, content
types, contest, options, problem, sessions, submission
Running migrations:
  No migrations to apply.
Django VERSION (1, 11, 4, 'final', 0)
User root exists, operation ignored
Django VERSION (1, 11, 4, 'final', 0)
2018-05-12 10:11:44,216 CRIT Supervisor running as root (no user
in config file)
Error: Another program is already listening on a port that one
of our HTTP servers is configured to use. Shut this program down
first before starting supervisord.
For help, use /usr/bin/supervisord -h
```

不用理会它, 继续就可以了;

4. 运行`exit`退出 Online Judge 的提示符。

大功告成, 这个时候可以刷新以下刚才在加载的界面, 您会惊喜的发现您的 Online Judge 能用了, 不再一直加载了。这个时候才是真正测试您的修改给整个系统带来了什么影响的时候。所以就愉快的调试起来吧。

## 4 保存镜像

调试完成后, 我们需要把镜像保存下来, 不用原来的一套镜像了 (如果机房晚上断电断网什么的 docker 容器可能会 gg, 需要重启)。方法很简单:

1. 运行`docker commit oj-backend`, 向源镜像提交更改。您会得到一串以 `sha256:` 开头的一串字符, 复制一下;
2. 用您喜欢的文本编辑器打开一键部署脚本目录下的 `docker-compose.yml` 文件, 把 `image:` 一行后的网址修改为刚才复制的一串字符, **要特别注意缩进!**, 当然也可以只复制前 32 个字符的缩略形式。

这样就完成了, 以后容器重新启动的时候也将使用您的镜像了。当然您也可以把这个镜像传到阿里云等等平台上, 然后把镜像替换成您在阿里云上的地址。