

[首页](#) [目录](#)[退出](#)

借助 Redux Thunk 实现异步 Action
Creator

React 组件连接
React Store 分发
Redux Action

前端导航栏体现用户
登录状态

PM2 部署项目

项目开发完毕，接下来就是把项目部署到真正的服务器上，部署工具使用 Node.js 社区大名鼎鼎的 **PM2**

服务器运行环境以及需要的软件包：

- Ubuntu v14.04
- Node.js v5.4.1 版本以上
- MongoDB v3.0 版本以上
- Nginx v1.4.6
- PM2 v2.1.5

安装 Nodejs

具体参考课程《Nodejs 乐高》的第二小节[安装 Nodejs](#)

安装 MongoDB

参考[Ubuntu 14.04](#)

安装 Nginx

```
sudo apt-get install nginx
```

Nginx 配置文件

配置一个域名，到 /etc/nginx/sites-enabled 目录下，
删除一个默认文件：

```
sudo rm default
```

然后，新建一个 client.conf 配置文件对应前端展示
页面的网址：

```
cd /etc/nginx/sites-enabled  
sudo touch client.conf
```

这个 client.conf 文件的内容如下：

```
server {
```

```

listen      80 default_server;

server_name your_domain_name;


location / {
    proxy_pass http://localhost:8000;
    proxy_set_header X-Forwarded-Proto $scheme;
    proxy_set_header Host $http_host;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_read_timeout 3m;
    proxy_send_timeout 3m;
}
}

```

用真实的域名替换 `your_domain_name`，在同一目录下，再添加对应后端 API 地址的配置文件 `server.conf`，文件内容如下：

```

server {
    listen      80;

    server_name your_subdomain_name.your_domain_name;


    location / {
        proxy_pass http://localhost:3000;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_set_header Host $http_host;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_read_timeout 3m;
        proxy_send_timeout 3m;
    }
}

```

```
}  
}
```

用真实的子域名替换 `your_subdomain_name`，域名替换 `your_domain_name`。最后，启动 Nginx，执行命令：

```
sudo service nginx start
```

若修改了 Nginx 的配置文件，需要执行下面命令，配置信息才能生效：

```
sudo service nginx reload
```

安装 PM2 包

登录到服务器，安装 PM2 包：

```
npm install -g pm2
```

添加 pm2 启动脚本

让 PM2 根据系统信息，自动生成启动脚本，先执行命令：

```
pm2 startup
```

当上面命令运行之后，会自动打印出类似如下一行命

令信息， `xxx` 会被您自己的登录用户名代替，直接复制这行命令，在命令行中粘贴运行就可以。

```
sudo su -c "env PATH=$PATH:/home/xxx/.nvm/v6.
```

当上面命令执行之后，在系统 `/etc/init.d/` 目录下，出现一个 `pm2-init.sh` 文件，即为 PM2 的启动脚本，当服务器重启的时候，会运行 `pm2-init.sh` 文件，从而启动 PM2。

详细文档参考 [启动脚本](#)

前端准备工作

在服务器的 Home 目录下，把项目从远端代码托管网站 GitHub 拽到服务器上的 `hand-in-hand-react-demo` 目录，执行命令：

```
git clone your_project_github_repo hand-in-ha
```

然后进入前端代码目录，安装 npm 包：

```
cd client  
npm install
```

修改配置文件 `src/settings.js`，更改 `host` 属性值：

```
export const Settings = {  
  host: your_api_domain_name  
}
```

用真实的后端 API 地址替换 `your_api_domain_name`

接下来打包项目，生成 `dist/bundle.js` 文件：

```
npm run build
```

创建目录 `public`，移动 `dist` 目录和 `index.html` 文件到 `public` 目录下：

```
mkdir public  
mv -rf dist index.html public
```

准备工作完成后，这时运行 `node server.js` 命令就能让前端代码跑起来了。但是当我们退出登录服务器的时候，`node server.js` 命令会终止运行，因此用 PM2 启动前端。

后端准备工作

进入后端代码目录，安装 npm 包

```
cd server  
npm install
```

复制 `config.default.js` 文件为 `config.js`：

```
cp config.default.js config.js
```

然后，修改 config.js 配置文件：

```
module.exports = {  
  port: 3000,  
  url: 'mongodb://localhost:27017/your_db_name',  
  secret: 'your_secret_key'  
};
```

用真实的生成 JWT Token 的密钥代替

your_secret_key，以及数据库的名称替换

your_db_name

这时运行 `node index.js` 命令，后端 API 服务就生效了。同样的问题，`node index.js` 不能长久运行，因此用 PM2 启动后端代码。

配置 JSON 文件，PM2 启动多个应用

现在，我们有前端和后端两个 Node.js 应用需要启动，所以在 Home 目录下，新建一个 `app.json` 文件，内容如下：

```
{  
  "apps": [{  
    "name": "api",  
    "script": "hand-in-hand-react-demo/server.js",  
  }, {  
    "name": "front-end",
```

```
    "script": "hand-in-hand-react-demo/client  
  }  
}
```

这个 `app.json` 文件将作为参数传递给 `pm2 start` 命令，如下：

```
pm2 start app.json
```

这样，前端和后端两个应用就都运行起来了。

欢迎添加 Peter 的微信：happypeter1983

冀ICP备15007992号-3