

项目创建

#软件安装

NODE

请先访问 <https://nodejs.org/zh-cn/>安装NODE（安装LTS版本即可）

淘宝镜像

使用[淘宝镜像 \(opens new window\)](#)可以快速安装NPM依赖包

```
1 npm install -g cnpm --registry=  
  https://registry.npm.taobao.org  
  npm install -g cnpm --registry=  
2
```

NODE

请先访问 <https://nodejs.org/zh-cn/>安装NODE（安装LTS版本即可）

淘宝镜像

使用[淘宝镜像 \(opens new window\)](#)可以快速安装NPM依赖包

```
1 npm install -g cnpm --registry=  
  https://registry.npm.taobao.org  
  npm install -g cnpm --registry=  
2
```

yarn

Yarn 是与 npm 类似的包管理工具，Yarn 缓存了每个下载过的包，所以再次使用时无需重复下载。同时利用并行下载以最大化资源利用率，因此安装速度更快。

安装 yarn

```
1 npm install -g yarn  
2
```

更多 yarn 的使用请访问[后盾人文档库\(opens new window\)](#)

#创建项目

vite

下面是使用 vite 构建基于 vue-ts 模板的项目 houdunren

```
1 yarn create vite houdunren --template vue-ts
2
```

#跨域请求

跨越请求的主要问题是携带 cookie，下面我们来解决 vue 的跨越请求问题。

以下示例环境为：后台houdunren.test，前台为 localhost:3000

#代理方式

如果后台要使用cookie 进行权限验证。这时就需要前台可以传递cookie，我们使用前台代理完成这个功能，使用这种方式后台不需要什么配置。

#axios

请求一般使用axios 发送，下面是对 url 的基本配置

不需要设置 axios 的 withCredentials 属性为 true

```
1 //请求拦截
2 instance.interceptors.request.use(
3   function (config) {
4     //如果请求不是以 / 开始时添加 /api 前缀
5     config.baseUrl = config.url[0] == '/' ? '' : '/api'
6     ...
7
```

#vite

下面是重点即在 vite.config.js 中配置代理，最终将实现通过 <http://houdunren.test/api>访问后台，解决跨越的问题。

```
1 export default defineConfig({
2   ...
3   //开发环境设置
4   server: {
5     proxy: {
6       //laravel 中获取 csrf-token 的接口
7       '/sanctum': {
```

```

8         //将/api访问转换为target
9         target: '
http://houdunren.test/sanctum
        target: '
10        changeOrigin: true,
11        rewrite: path => path.replace(/^\/sanctum/, ''),
12    },
13    '/api': {
14        //将/api访问转换为target
15        target: '
http://houdunren.test/api
        target: '
16        //跨域请求携带cookie
17        changeOrigin: true,
18        //url 重写删除`/api`
19        rewrite: path => path.replace(/^\/api/, ''),
20    },
21 },
22 },
23 })
24
25

```

#laravel

因为上面是使用代理解决跨域问题，所以如果后台使用 laravel 不需要进行配置。下面对几个文件进行说明

.env

修改 `.env` 来对域名等进行配置

```

1 APP_URL=
http://houdunren.test
APP_URL=
2

```

config/cors.php

因为上面使用代理操作，所以需要把 `supports_credentials` 设置为 `false`

```

1 'paths' => ['api/*', 'sanctum/csrf-cookie'],
2

```

```

3  'allowed_methods' => ['*'],
4
5  'allowed_origins' => ['*'],
6
7  'allowed_origins_patterns' => [],
8
9  'allowed_headers' => ['*'],
10
11 'exposed_headers' => [],
12
13 'max_age' => 0,
14
15 'supports_credentials' => false,
16

```

config/sanctum.php

因为前台使用 `localhost:3000` 所以要保证该域名存在

```

1  ...
2  'stateful' => explode(',', env('SANCTUM_STATEFUL_DOMAINS', sprintf(
3      '%s%s',
4      'localhost,localhost:3000,127.0.0.1,127.0.0.1:8000,::1',
5      env('APP_URL') ? ',' . parse_url(env('APP_URL'), PHP_URL_HOST) : ''
6  ))),
7  ...
8

```

#csrf-token

因为使用 cookie 请求 laravel 所以后台会进行 csrf 验证，需要先获取 csrf-token 的 cookie。axios 等框架会自动携带这个 cookie 完成后台的 csrf 验证。

下面是登录操作的代码

```

1  ...
2  const login = async () => {
3      axios.get('/sanctum/csrf-cookie').then(async response => {
4          await axios.login(form)
5          ...
6      })

```

```
7 }  
8 ...  
9
```

下面是 loginController 控制器文件，我们以 `api/login` 路由访问。需要使用 **web** 的 **guard** 完成 session 登录。

```
1 ...  
2 Auth::guard('web')->login($user);  
3 ...  
4
```