# Web Common

前端公共仓库，包含文档说明、公共库函数、公共组件、构建脚本等，外部项目以SVN外部链接、Git子模块形式依赖

SVN地址：<http://192.168.0.22/svn/web-common/>

# 项目规范

参考`./项目规范.md`文档

# 公共组件

公共组件目前没有在多项目中进行复用，主要有三个基于EWTP项目的组件：Table、Form、Dialog

## 共用函数

./src/utils/http: 网络请求库接口

# 配置文件

**./.env**

Vue CLI和React Scripts都定义了相同的环境变量文件，它们的处理和使用方式基本一致，按照模式和是否被版本管理工具可以建立多个文件，`./.env`文件会在所有模式下载入，`./ .env.development.local`仅在本地开发模式起效，同时会被版本管理工具排除，所以用于本地开发根据不同需求配置环境变量，比如配置API代理主机、BASE\_URL等

环境变量使用也很简单，单行形式key=value格式

如果环境变量需要传递到前端页面，需要按照特定格式才会被脚手架处理，Vue项目按照`VUE\_APP\_xxx`格式，React项目按照`REACT\_APP\_xxx`格式

**./src/env.js**

为了方便前端代码管理环境变量，定义env.js文件集中处理后端传递过来的环境变量，同时针对Vue/React统一了读取设置环境变量的方法

默认下处理两个环境变量

PUBLIC\_PATH：webpack public path，可用于配置CDN发布地址

BASE\_URL：http request baseURL，可用于配置本地直接调用远端接口而不用走后端代理，需要后端配置CORS允许跨域

# 构建文件

* 框架config文件，主要针对脚手架默认配置进行覆盖或者添加自定义配置比如CSS模块生产环境自定义命名格式，扩展后端API代理模式等

1. Vue框架，`./build/vue.config.js`
2. React框架，`/build/react/ config-overrides.js`

* Lint，模板项目依赖gs-lint包

1. ESLint JS/TS文件格式检查

Vue项目：`/build/eslintrc.js`，模板项目依赖` gs-lint/eslint/eslint-vue`

React项目：`./build/react/eslintrc.js`，模板项目依赖` gs-lint/eslint/eslint-react`

1. StyleLint样式文件格式检查，模板项目依赖` gs-lint/stylelint/.stylelintrc`

`./build/stylelintrc.js`

* API Proxy

`./build/proxycfg.js`

默认代理path为`^/api`，代理主机：<http://192.168.0.203:8082>

可通过环境变量`PROXY\_HOST`指定实际主机，如果需要修改代理path指定`dftApiPrefix`

手机屏幕的截图

描述已自动生成

屏幕上有字

描述已自动生成

* Build

项目构建默认输出到`./dist/${outputName}`目录下

* Pack

pack会把构建输出目录打包成单个tgz格式文件便于文件传输和部署，同时会在打包前生成一个meta.json元文件包含项目版本信息等数据。具体参考文件`./build/pack.js`

# 项目模板

根据公司实际项目背景，基于EWTP项目封装了Vue + Element+ Vue CLI项目模板便于快速搭建项目框架，目前项目模板以zip格式打包存储，将模板文件拷贝到项目目录解压出来进行对应修改即可快速搭建项目框架，后期可以考虑以npm包形式发布，提供cli命令来生成项目

模板文件解压出来后参考当前目录下` interpolate.md`文件修改几处地方，然后配置依赖web common，安装package.json依赖，完成后就可以将项目运行起来

Vue项目模板：`./build/template/vue/ template.zip`

模板文件说明

./src/env.js

基于web common中env.js在特定项目处理环境变量

./src/debug.js

针对环境变量所定义的本地调试文件，这里面配置的环境变量值优先级最高，因为修改.env文件中的环境变量需要重启服务器，所以一般情况下直接在调试文件中直接修改修改环境变量值最方便，同时浏览器也会自动刷新