

前端工程说明

0 特性

nodejs驱动开发的前端工程具有以下特征：

- 1, 与接口服务器的交互和数据模拟
- 2, css代码的预编译和后编译,不用再手动处理兼容
- 3, html模板功能
- 4, js模块化
- 5, 能够根据开发，测试，生产环境弹性打包
- 6, 可以对本地的图片进行压缩和base64编码处理

nodejs驱动前端工程不但可以加快开发速度，还可以增强代码的可维护性和健壮性，并可以进行一定程度的优化。

1 安装

1.1 nodejs

去 [官方网站](#) 下载node安装包，推荐下载LTS版本，最新版亦可。

注：如果你本机已经安装过nodejs，请确保版本号 > 6.0，npm版本号 > 3

1.2 全局依赖

1.2.1 nrm

npm默认是使用位于国外的官方源下载，速度较慢，所以需要使用 `nrm` 把安装源切换到国内。

```
// 安装nrm
sudo npm i -g nrm

// 查看全部可用源
nrm ls

// 使用淘宝源
nrm use taobao

// 源测速
nrm test
```

1.2.2 yarn

`yarn` 是Facebook推出的新一代nodejs包管理工具，相比 `npm` 来说带来了本地包缓存，包依赖版本锁定2大变化。

```
// 安装yarn
sudo npm i -g yarn

// 在项目下安装全部依赖包(按照package.json)
yarn

// 在项目下增加包
yarn add yourPack

// 在项目下删除包
yarn remove yourPack
```

2 开发

2.1 目录结构说明

```
~/yourpath/FE_PROJECT
├─ common ----- 前端公共库
├─ node_modules ----- 依赖包代码
├─ package.json ----- 依赖包目录
├─ pages ----- 子工程目录
│   ├── greenBee
│   ├── mars
│   ├── star
│   ├── wechat
│   ├── xinhua
│   ├── xinhua2
│   └─ zhuowang
├─ page.config.js ----- 编译目录配置文件
├─ prod.config.js ----- 生产环境配置文件
├─ README.md ----- 本说明文档
├─ test.config.js ----- 测试环境配置文件
├─ webpack.config.js ----- 开发环境配置文件
└─ yarn.lock ----- yarn锁定依赖目录
```

2.2 编译配置文件说明（page.config.js）

```

const path = require('path');
const glob = require('glob');
const DIR_NAME = 'greenBee'; // 当前编译的文件夹名字

let getEntries = function(pattern){
  let entries = {};
  glob.sync(`./pages/${DIR_NAME}/${pattern}` ).forEach(item=>{
    entries[item.split('/').pop().split('.').shift()] = item;
  })
  return entries;
}

module.exports = {
  dirname: DIR_NAME,
  designWidth: 640, // 设计稿宽度
  useRem: true, // 是否使用rem来布局
  /*
   * 入口js文件
   * js文件夹下的第一级.js后缀文件会被匹配到
   * 不做为入口的js文件请放入文件夹中，如 js/util/
   *
   */
  entry: getEntries('js/*.js'),
  /*
   * 模板html文件
   * dirname文件夹下的第一级.html后缀文件会被匹配到
   * 不做为模板的html文件请放入文件夹中，如 /layout
   *
   */
  template: getEntries('*.html'),
  resolve: { // 全局别名
    alias: {...},
  },
  proxy: { // 接口转发
    '/remote_api': {
      target: 'http://192.168.2.183:8080',
      secure: false,
      changeOrigin: true,
      pathRewrite: {
        '^/remote_api' : ''
      },
    },
    ...
  }
};

```

2.3 主机及端口号说明

为了方便移动端开发，主机启动的是 `0.0.0.0`，本机ip会自动获取

```
~/yourpath: vi webpack.config.js

// 修改publicPath
module.exports = {
  ...,
  output: {
    publicPath: 'http://本机IP:8088/',
    ...
  },
  ...
}
```

2.4 开发

执行以下命令启动开发服务器：

```
// 进入前端工程目录
cd FE_PROJECT

// 安装依赖包
~/yourpath/FE_PROJECT: yarn

// 启动开发服务器
~/yourpath/FE_PROJECT: yarn dev
```

以上命令启动一个端口为8088的nodejs服务器用于开发。打开浏览器输入 <http://本机IP:8088/> 即可看到页面。

该页面默认加载了 `微信调试工具(vConsole)`，并会在js文件发生变化时自动刷新浏览器。

2.5 测试和生产

`yarn test` 命令会在mobile文件夹下面生成一个test文件夹，里面存放了当前开发项目的静态资源。

```
// 测试目录文件夹结构
~/yourpath/mobile/test: tree

├─ ...
├─ ...
├─ your_project ----- 打包后的资源文件夹
│   ├── css
│   ├── greenBee_focus.html
│   ├── greenBee_form.html
│   ├── greenBee_index.html
│   ├── images
│   └─ js
├─ ...
└─ ...
```

`yarn build` 命令会在mobile下生成一个以当前项目命名的文件夹，同样存放了静态资源。

```
// 生产目录文件夹结构
~/yourpath/mobile: tree

├─ ...
├─ ...
├─ your_project ----- 打包后的资源文件夹
│   ├── css
│   ├── greenBee_focus.html
│   ├── greenBee_form.html
│   ├── greenBee_index.html
│   ├── images
│   └─ js
├─ ...
├─ ...
└─ templates ----- 最终html模板
```

注:

- 1, 这两个命令生成的html和css中的静态资源, 均指向到 `/mobile` 目录下。
 - 2, test 会对css和js资源进行压缩。
 - 3, build 会对css/js/png图片进行压缩和优化, 并去除 `vConsole` 调试工具。
 - 4, 均未对html和jpg图片资源进行压缩。
- todo: 添加线上调试配置功能, 用户选择 "线上调试后", 加载 `vConsole` 调试工具。

2.6 重要依赖包

本项目中主要用到的开发依赖为:

- 1, [webpack](#) 用于对js, css资源模块进行打包。
- 2, [webpack-dev-server](#) 用于启动开发服务器, 该服务器可以做自动刷新, 接口转发
- 3, [ejs-html-loader](#) 把html文件作为ejs模板, 可以实现变量注入和模板插入。