# Webpack最佳实践

什么是前后端分离？

目前业界比较有代表性的前后端分离的例子是SPA(Single-page application)，所有用到的展现数据都是后端通过异步接口(AJAX/JSONP/WEBSOCKET)的方式提供的。

webpack本身的定位是一个资源管理和打包构建工作，本身的强大之处在于对各种静态资源的依赖分析和预编译，在实际开发中官方还推荐了一个快速读取webpack配置的server环境**webpack-dev-server，**一个适用于开发环境的，基于express + webpack-dev-middleware实现的，支持实时更新，内存构建资源的开发服务器，通过简单的配置即可满足webpack开发环境中的一系列需求。

**借助node和http-proxy实现跨域通信**

**将webpack配置和node server进程打通**

1. 解耦webpack中的配置

对配置文件进行重构，将之前的webpack配置文件拆解成了webpack.config.js，project.config.js和environments.config.js三个文件，三个文件各司其职，又可互相协作，减少维护成本

1. 整合webpack在开发环境中依赖的中间件

参考webapck-dev-server中的实现，我们可以将webpack-dev-middleware和webpack-hot-middleware加入到我们的express配置中，这样当我们执行下述的时候，就能一键完成webpack基础配置，热更新以及epxress服务的启动，并且可以完全根据express的配置说明来自定义扩展我们的前端开发资源。

**前端资源路径设计**

* html：html页面，结合到服务后一般称为模板资源，是所有资源的入口和结果呈现页；
* js：javascript执行脚本资源，基于现代Javascript框架开发后通常还需要借助babel，typescript等进行编译处理，分为build前后build后两套代码；
* css：样式资源，如果采用了less，sass等工具处理后会也会从.less和.sass编译成.css文件;
* static: 静态资源，通常会包含有font，image，audio，video等静态文件，结合到服务框架中一般需要设定特定的访问路径，直接读取文件加载。
* 在wepback的配置中，前端资源路径我们通常是借助path和publicPath 对构建出来的前端资源进行索引，由于webpack采用了基于内存构建的方式，path通常用来用来存放打包后文件的输出目录，publicPath则用来指定资源文件引用的虚拟目录。

**mock数据模拟**

1. 配置专属的mock路由模块 我们可以在我们的server项目下的routes模块中加入一个mock模块
2. 搭建独立的mock服务 如果企业中有部署独立的mock服务器，如puer+mock：我们也可以通过修改简单的proxy.config来直接实现需要mock的请求地址转发，相对修改就比较简单

**预览打包后的资源效果**

1. 找到构建资源生成目录, 确认构建资源已存在：
2. 修改express的文本配置信息，加入构建完成后的静态资源地址配置：
3. 启动预览页面，访问:localhost:3000即可

工程演示demo（<https://github.com/taikongfeizhu/webpack-develop-startkit>）

参考文献：

[**http://music.163.com/song/27646786?userid=419944244**](http://music.163.com/song/27646786?userid=419944244)