

---

# 爱旅行项目—前端架构设计

## 1.1 分层架构&模块化

谈到框架设计，很多人都会想到 MVC、MVP、MVVM 等，那我们前端开发为什么要进行框架设计那？使用原始的进行开发有什么问题吗？首先我们从这几点出发看一下原始的开发是否存在以下问题：

- 难以维护
- 加载缓慢
- 体验差
- 重复编码
- 扩展困难
- 前后端耦合度高

我们在原始开发的时候都会遇到以上的问题，因此从这些问题上我们发现使用前端框架的重要性，有些人或多或少的接触过框架，但是独对前端框架的认知还不够完善，我们爱旅行这个项目使用**分层架构设计**（把功能相似，抽象级别相近的实现进行分层，使逻辑变得清晰，容易理解和维护，也被称作为多层架构或 N 层架构），在这里我们使用的是类 MVVM 的分层架构方式，但还不完全是 MVVM 的分层架构方式；分层架构的主要优势在于：易维护、可扩展、易复用、灵活性高等特点深受前端开发工程师喜爱；

在使用分层架构的同时我们还需要使用比较重要的技术：**模块化**——是指解决一个复杂问题时自顶向下逐层把系统划分成若干模块的过程，有多种属性，分别反映其内部特性。同时模块化还可以解耦实现并行开发。主要的模块化解方案有：AMD（requirejs）、CMD（seajs）、CommonJS、ES6。模块化用来分割，组织和打包软件。每个模块完成一个特定的子功能，所有的模块按某种方法组装起来，成为一个整体，完成整个系统所要求的功能。在系统的结构中，模块是可组合、分解和更换的单元。模块化是一种处理复杂系统分解成为更好的可管理模块的方式。它可以通过在不同组件设定不同的功能，把一个问题分解成多个小的独立、互相作用的组件，来处理复杂、大型的软件。

关于模块化的技术有很多，我们上面也列举了四种，我们在这里只使用一种方式进行开发，为了保证技术的前瞻性，我们在这里使用 ES6 进行代码的开发；虽然现在大部分浏

览器还不完全支持 ES6，但是 ES6 在企业中应用非常广泛；图 1 框架设计图，可参照此图理解：

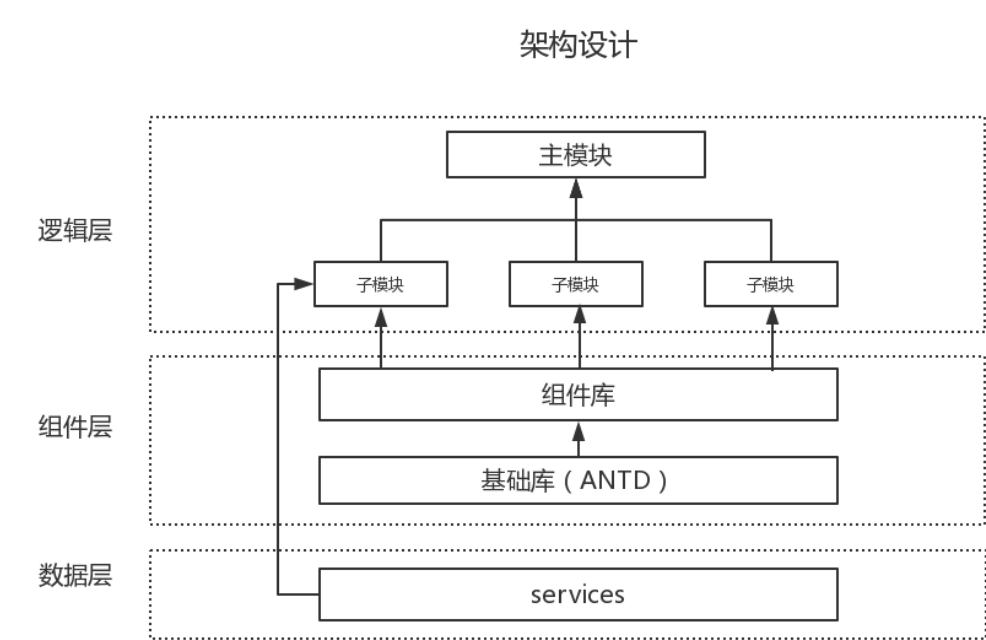


图 1

## 1.2 软件过程

由于我们这个项目的工作量比较大，时间比较短，需求会有变化因此我们采用敏捷开发的方式，实现快速迭代开发。敏捷开发既满足不断变化的优点又避免了人员的浪费，但是敏捷开发对成员的能力要求比较高，所以要提高我们的个人能力。关于敏捷开发可参照《硝烟中的 Scrum 和 XP》、《敏捷软件开发》这里就不对敏捷开发进行过多的讲解了。

## 1.3 分离方式

关于分离方式我在这里要多说一下，分离方式主要有三种：

- 不分离
- 部分分离
- 完全分离

在我们以前的项目开发中，采用的是不分离或者部分分离进行开发，不分离和部分分离有什么缺点呢？不分离：前后端共用同一个项目目录，本地开发环境搭建成本高，项目比较复杂不宜维护且维护成本高，发布风险高，不利于问题的定位和修改。部分分离：本地环境搭建成本较高，需要后端提供页面模板（JSP 等），更新和修改模板都需要后端人员操作，效率低且不易维护，发布方式需要同时发布，且沟通成本比较高；通过以上两点我们可以看出集成部署的缺点是存在的，但安全性比较高；有没有一种技术技能保证安全性又能解决以上的问

题呢？当然是有的，那就是完全分离即分布式，完全分离又分为两种：分离开发集成部署，分离开发分离部署。我们这里是用第二种：分离开发分离部署，前端我们使用纯的 HTML 通过接口的方式进行数据的交互，降低系统的复杂度，部署时单独部署到一台服务器上使用代理进行数据的交互；关于部署和本地测试请参照《前端打包部署》、《前端项目本地调试》两个文档进行学习，关于接口数据的使用和模拟请参照《Rap 模拟数据接口》文档进行学习

## 1.4 框架

在进行框架选择上我们要注意这样几个问题：一、项目需要的浏览器支持；二、是否有满足需求的组件库；三、框架的稳定性和成熟度；四、扩展的容易度；五、学习成本；六、后期维护成本等；我们需要从诸多因素中进行考虑，保证我们选择的框架是能够被支持的不能因为选择的失误导致我们更换框架，这样是得不偿失的，毕竟框架是最底层的也是最重要的，就像盖楼没有地基一样，没有地基最终会倒。且盖得越高损失的也是最惨。因此框架的选择是至关重要的；

在这里我们要使用现代技术的发展潮流与方向同时提高个人的能力，在这里就没有考虑第一条：项目需要的浏览器支持。那我们这里使用什么框架进行开发呢？在这里我们使用 React 框架进行后期项目的开发。为什么选择 React 进行项目开发呢，首先第一条就是 React 是时下最流行的前端框架之一，也是现在企业级开发中比较常用的技术，关于 React 的技术讲解详见《前端-React 技术分析》文档，此文档会详细讲解 React 的相关技术及实现原理等信息；其次 React 有非常完善的第三方组件库 ANTD，此组件库可以满足我们正常的业务开发，且文档非常齐全；综上我们选择使用 React 框架结合 CSS3 进行项目开发。

## 1.5 构建工具

构建工具我们使用 webpack 进行项目的构建和部署，Webpack 可以将多种静态资源 js、css、less 转换成一个静态的文件（如图 2），减少页面的请求，同时也减少了我们去转义 less 或 ES6 语法等工作，大大的提高了我们的开发效率；想了解更多关于 webpack 的信息请参照：《webpack 使用指南》，此文档里面会详细讲解如何使用 webpack 进行构建和部署的；

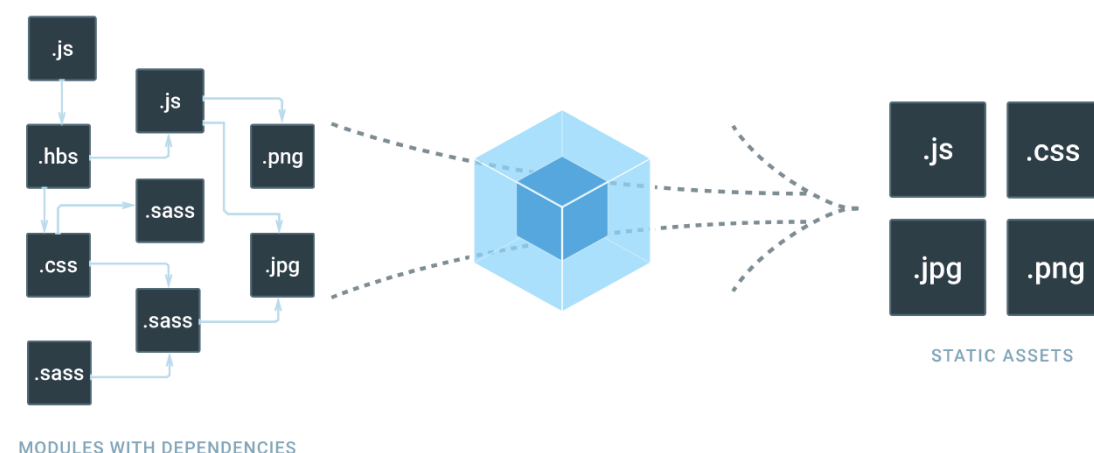


图 2

---

## 1.6 版本控制

版本控制我们使用 `git` 进行管理，使用 OSChina 进行代码版本的管理，OSChina 的地址：<https://git.oschina.net/>；关于 `git` 的使用这里就不做过多的讲解，常用的命令有：`git clone`、`git add .`、`git commit -m 'desc'`、`git pull`、`git push` 等；

## 1.7 开发工具

现在的开发工具有很多，`webstrom`、`sublime`、`vscode` 等，我们这里使用 `vscode` 进行代码的开发，`vscode` 的下载地址：<https://code.visualstudio.com/Download> 如果大家之前使用 `webstrom` 比较熟悉，可以在 `vscode` 中导入 `webstrom` 的快捷键可在插件中搜索：`IntelliJ IDEA Key Bindings` 安装完成之后重启即可；`vscode` 的特点是程序相对较小且功能比较齐全，编译效率高；因此我们在此项目开发中使用 `vscode` 作为项目开发的编辑器；

## 1.8 发布方式

关于发布方式我们使用代码提取->打包->部署到线上服务器，具体关于如何打包部署的请参照《前端打包部署》文档进行学习；

