

Web前端工程师知识扩展

什么是单页面应用（SPA）

单页面应用是仅包含单个网页的应用，目的是为了提供类似于本地应用的流畅用户体验。

要实现单页面应用，现在已经有很多现成的框架了，比如Angular、React、Vue、Ember.js、Backbone.js等等。它们都是很全面的开发平台，为单页面应用开发提供了必需的页面模板、路径解析和处理、后台服务api访问、DOM操作等功能。

单页面应用(SPA)的优缺点

优点：

1. 用户体验好、快，内容的改变不需要重新加载整个页面，避免了不必要的跳转和重复渲染。
2. 前后端职责分离，架构清晰：前端进行交互逻辑，后端负责数据处理。
3. 基于上面一点，SPA相对对服务器压力小。

缺点：

1. seo 不利于搜索引擎优化
2. 初次加载页面更耗时
3. 前进、后退、地址栏等，需要程序进行管理；
4. 书签，需要程序来提供支持；

什么是MVC架构

MVC是软件工程中的一种软件架构模式，MVC把软件系统分成三个基本部分：

- 模型（Model）
- 视图（View）
- 控制器（Controller）。

模型（Model） “数据模型”（Model）用于封装与应用程序的业务逻辑相关的数据以及对数据的处理方法。“模型”有对数据直接访问的权力，例如对数据库的访问。“模型”不依赖“视图”和“控制器”，也就是说，模型不关心它会被如何显示或是如何被操作。但是模型中数据的变化一般会通过一种刷新机制被公布。为了实现这种机制，那些用于监视此模型的视图必须事先在此模型上注册，从而，视图可以了解在数据模型上发生的改变。

视图（View） 视图层能够实现数据有目的的显示（理论上，这不是必需的）。在视图中一般没有程序上的逻辑。为了实现视图上的刷新功能，视图需要访问它监视的数据模型（Model），因此应该事先在被它监视的数据那里注册。

控制器（Controller） 控制器起到不同层面间的组织作用，用于控制应用程序的流程。它处理事件并作出响应。“事件”包括用户的行为和数据模型上的改变。

前端工程化

凡是重复的，必须使用工具自动完成。

工具众多，我们就有一种想法，能不能有一种工具能帮我们自动生成雪碧图、css压缩、图片压缩等等，然后就出现了前端工程化。

前端工程化一般可分为五个步骤：

1. 初始，生成基础目录结构和样式库。
2. 开发，实时预览、预编译。
3. 构建，预编译、合并、压缩。
4. 发布，将构建后静态文件发布上线。
5. 打包，资源路径转换，源码打包。

什么是组件化开发？

1. 页面上的每个独立的可视/可交互区域视为一个组件；
2. 每个组件对应一个工程目录，组件所需的各种资源都在这个目录下就近维护；
3. 每个组件相对独立，页面只不过是组件的容器，组件自由组合形成功能完整的界面；
4. 当不需要某个组件，或者想要替换组件时，可以整个目录删除/替换。

组件化开发有如下的好处：

-
1. 降低整个系统的耦合度，在保持接口不变的情况下，我们可以替换不同的组件快速完成需求，例如输入框，可以替换为日历、时间、范围等组件作具体的实现。
 2. 调试方便，由于整个系统是通过组件组合起来的，在出现问题的时候，可以用排除法直接移除组件，或者根据报错的组件快速定位问题，之所以能够快速定位，是因为每个组件之间低耦合，职责单一，所以逻辑会比分析整个系统要简单。
 3. 提高可维护性，由于每个组件的职责单一，并且组件在系统中是被复用的，所以对代码进行优化可获得系统的整体升级。例如某个组件负责处理异步请求，与业务无关，我们添加缓存机制，序列化兼容，编码修正等功能，一来整个系统中的每个使用到这个组件的模块都会受惠；二来可以使这个组件更具健壮性。

在团队开发中，组件化带来的优势是便于协同开发，由于代码中的耦合度降低了，每个模块都可以分拆为一个组件，例如异步请求组件，路由组件，各个视图组件。团队中每个人发挥所长维护各自组件，对整个应用来说是精细的打磨。