

(4) 前端进阶1

一 前端工作流

1

- 前端测试
- 作为一个项目负责人怎么协调多人协作
- 接口文档的制定（给自己挖了一个坑）
- 需求不明确，接口文档是不是越详细越好
- webpack和gulp区别（模块化与流的区别）

2 webpack【“前端模块化打包机”】和 gulp【“流水线、按顺序执行task”】的区别

2.1 gulp强调的是 前端开发的工作流程【“偏向于流水线的操作”】，我们可以通过配置一系列的 任务task，定义 task 处理的事务（例如 文件压缩合并、雪碧图、启动 server、版本控制等），并定义他们的执行顺序，从而使得 gulp 按照我们预期的顺序去执行这些 task、从而构建项目的整个前端开发流程。

webpack 是一个 前端模块化方案、更侧重于模块的大包。

我们可以把开发中的所有资源（图片、JS、CSS文件等）都是为一个模块。

通过 loader（加载器）和 plugins（插件）对资源进行处理，打包成符合生产环境部署的前端资源！！

gulp强调的是前端开发的工作流程，我们可以通过配置一系列的task，定义task处理的事务（例如文件压缩合并、雪碧图、启动server、版本控制等），然后定义执行顺序，来让gulp执行这些task，从而构建项目的整个前端开发流程。

webpack是一个前端模块化方案，更侧重模块打包，我们可以把开发中的所有资源（图片、js文件、css文件等）都看成模块，通过loader（加载器）和plugins（插件）对资源进行处理，打包成符合生产环境部署的前端资源。

二 流行的框架

1 redux的作用：

本质：redux就相当于整个应用的管家。“redux类似于该应用的一个全局变量、共所有的组件去使用！！！”

在组件化的应用中，会有着大量的组件层级关系，深嵌套的组件与浅层父组件进行数据交互，变得十分繁琐困难。而redux，站在一个服务级别的角度，可以毫无阻碍地将应用的状态传递到每一个层级的组件中。redux就相当于整个应用的管家。

2 redux的常用方法【2属性：state、listeners；3方法：getState、dispatch（action）、

subscribe (listener) 】

提供 `getState()` 方法获取 state;

提供 `dispatch(action)` 方法更新 state;

通过 `subscribe(listener)` 注册监听器;

等等

3 react 对比 angular

React对比Angular是思想上的转变，它也并不是一个库，是一种开发理念，组件化，分治的管理，数据与view的一体化。它只有一个中心,发出状态，渲染view，对于虚拟dom它并没有提高渲染页面的性能，它提供更多是利用jsx便捷生成dom元素，利用组件概念进行分治管理页面每个部分(例如 header section footer slider)

4 vue双向绑定的原理

核心原理：

通过数据劫持结合 发布者-订阅 模式的方式来实现的。

利用 `Object.defineProperty()` 这个方法重新定义了对象获取 属性值 (get) 和 设置属性值 (set) 。

vue数据双向绑定是通过数据劫持结合发布者-订阅者模式的方式来实现的。利用了 `Object.defineProperty()` 这个方法重新定义了对象获取属性值(get)和设置属性值(set)。

5 说说vue react angular jquery 的区别

JQuery与另外几者最大的区别是，JQuery是事件驱动，其他两者是数据驱动。

JQuery业务逻辑和UI更改该混在一起，UI里面还参杂这交互逻辑，让本来混乱的逻辑更加混乱。

Angular，vue是双向绑定，而React不是

其他还有设计理念上的区别等

三 Node.js (问得挺多的，应该好好总结一下)

1 node的事件 【emitter. addListener | removeListener | emit | on】方法？？

`emitter.addListener(eventName, listener)`，`emitter.emit(eventName[, ...args])`，`emitter.on(eventName, listener)`，`emitter.removeListener(eventName, listener)`等

2 node的特性。 适合处理什么场景??

Node借助 事件驱动【“天生异步??”】，非阻塞 I/O 模型变得轻量 和 高效，非常适合运行在 分布式设备的 数据密集型 实时应用!!

Node.js借助事件驱动，非阻塞I/O模型变得轻量和高效，非常适合运行在分布式设备的数据密集型实时应用。

3 讲讲 Express

Express 是一个简洁而灵活的 node.js Web应用框架, 提供了一系列强大特性帮助你创建各种 Web 应用，和丰富的 HTTP 工具。

1 4 promise的状态有哪些??

等待 (pending)、已完成 (fulfilled)、已拒绝 (rejected)

wa b