

面筋系列①——滴滴sp一面面试真题

滴滴sp前端一面



落神-滴滴出行实习生

编辑于 2020-08-06 09:36:19

@稀土掘金技术社区

前言

最近在看一些面经，去看看别人面试遇到的题目，然后每看一篇面经就整理一次题目及答案。（之后大概率会持续更新）

关于本文

本文是因为最近投了个滴滴的前端实习岗位，所以在牛客查了查有关滴滴的面经，于是乎看到这个 [滴滴sp前端一面](#)，整理了一下关于里面题目的答案。当然啦，如果哪里讲得不好的或是讲错的，还是希望大家评论纠正一下我的错误。（涉及到React的题目暂时不讲解，作者时间安排还没学React>_<）

滴滴sp一面题目及解答

1. 手写发布订阅模式

直接第一题我就懵了，想不明白直接到掘金查了查发布订阅模式，原来是有关设计模式的知识。这一块的知识缺了，那就补，因为我还没搞懂这里暂且就不详解了。

2. Vue的双向绑定原理

Vue.js是采用数据劫持结合发布者-订阅者模式的方式，通过 `Object.defineProperty()` 来劫持各个属性的setter、getter，在数据变动时发布消息给订阅者，触发相应的监听回调。

- ① 实现数据监听器Observer，能够对数据对象的所有属性进行监听，如有变动可拿到最新值并通知订阅者
- ② 实现一个指令解析器Compile
- ③ 实现一个Watcher，作为Observer 和 Compile 的桥梁，能够订阅并收到每个属性变动的通知，执行指令绑定的相应回调函数，从而更新视图
- ④ Mvvm入口函数，整合以上三者

3. Vue和React的区别

「相同点：」

- ① 都使用 `Virtual Dom`
- ② 都使用组件化思想，流程基本一致
- ③ 都有成熟的社区，都支持服务端渲染

「不同点：」

- ① 组件通信不同
- ② 模板渲染方式不同
- ③ Diff算法不同
- ④ 事件机制不同
- ⑤ 监听数据变化的实现原理不同

(这里希望大家能帮我补充一下)

4. Webpack了解过吗，讲一下怎么做的？loder和plugin的区别

(1) 是什么？

- Webpack是一个模块打包工具，可以使用它管理项目中的模块依赖，并编译输出模块所需的静态文件。它可以很好地管理、打包开发中所用到的HTML、CSS、JS和静态文件等，让开发更高效。对于不同类型的依赖，webpack有对应的模块加载器，而且会分析模块依赖间的依赖关系，最后合并生成优化的静态资源。

(2) 怎么做的？（基本功能、工作原理）

- 代码转换、文件优化、代码分割、模块合并、自动刷新、代码校验、自动发布

(3) loader 和 plugin的区别

- **「loader」** 是使webpack拥有加载和解析非js文件的能力
- **「plugin」** 可以拓展webpack的功能，使得webpack更加灵活。可以在构建的过程中通过webpack的api改变输出的结果

5. Axios了解吗？Jsonp原理是什么？

(1) Axios是什么？

- axios是一个基于promise的HTTP库，可以用在浏览器和node.js中

(2) 「Axios有什么特性」？（拓展）

- 从浏览器中创建 **XMLHttpRequests**
- 从node.js创建http请求
- 支持Promise Api
- 拦截请求和响应
- 可以取消请求
- 自动转换JSON数据
- 客户端支持防御XSRF

(3) Jsonp是什么？（Jsonp原理）

- 通过动态创建script标签，其src指向非同源的url，并传递一个callback参数作为函数名的函数的调用和一系列参数，页面接收到响应后执行回调并对数据进行处理

6. 手写ajax，readyState=1、2、3代表什么？

(1) 手写ajax（简易版）

```
function ajax ( url , successFn ){

    const xhr = new XMLHttpRequest()
    xhr.open( 'GET' , url , true)

    xhr.onreadystatechange = function() {
        if( xhr.readyState == 4 ){
            if( xhr.status == 200 ){
                successFn(xhr.responseText)
            }
        }
    }

    xhr.send(null)
}
```

(2) readyState 是什么? readyState=1、2、3代表什么?

- 是HTTP请求的状态，当一个XMLHttpRequest初次创建时，这个属性的值从0开始，直到接收到完整的HTTP响应，这个值增加到4
- 0: (「未初始化」) 对象已建立，但是尚未初始化（尚未调用到open方法）
- 1: (「初始化」) 对象已建立，尚未调用 `send` 方法
- 2: (「发送数据」) `send` 方法已调用，但是当前的状态及http头未知
- 3: (「数据传送中」) 已接收部分数据，因为响应及http头不全，这时通过 `responseBody` 和 `responseText` 获取部分数据会出现错误
- 4: (「完成」) 数据接收完毕，此时可以通过 `responseXml` 和 `responseText` 获取完整的响应数据

7. 讲讲深拷贝和浅拷贝

这里参考 [聊聊对象深拷贝和浅拷贝](#)

浅拷贝：创建一个新对象，这个对象有着原始对象属性值的一份精确拷贝。如果属性是基本类型，拷贝的就是基本类型的值，如果属性是引用类型，拷贝的就是内存地址，所以如果其中一个对象改变了这个地址，就会影响到另一个对象

「浅拷贝方法」

- `Object.assign`
- 拓展运算符...
- `Array.prototype.slice`

- `Array.prototype.concat`

深拷贝：将一个对象从内存中完整的拷贝一份出来，从堆内存中开辟一个新的区域存放新对象，且修改新对象不会影响原对象

「深拷贝方法」

- `JSON.stringify()`
- 手写深拷贝

8. 手写深拷贝

老题目直接上代码了

scss 复制代码

```
function deepClone( obj = { } ){

    if( typeof obj !== 'object' || obj == null ){
        return obj
    }

    let result
    if( obj instanceof Array ){
        result = [ ]
    } else {
        result = { }
    }

    for( let key in obj ){
        if( obj.hasOwnProperty(key) ){
            result[key] = deepClone( obj[key] )
        }
    }

    return result
}
```

9. setTimeout是怎么实现的？放入到队列中吗？代码什么时候放入？

这题我本来也就想着是一个简单的有关任务队列、讲一下Event loop的题目，但是我看了一下这篇面经主人写的回答是这样的

不是用队列实现的，是用hash对象实现的，是代码遇到时才放入

--[滴滴sp前端一面](#)

”

我一下子也没想明白是啥意思，我搜了一下相关的知识，也搜不到，不知道有没有大佬能给我科普一下

10. 箭头函数和普通函数有什么区别

「箭头函数区别：」

- ① 不能通过 new 关键字调用, 没有原型
- ② 不可以改变 this 绑定，由外层非箭头函数决定，所以使用 call, apply, bind 也不会影响
- ③ 不支持 arguments，所以根据作用域链，会拿到外层函数的 arguments
- ④ 不支持重复命名参数
- ⑤ 隐式返回

11. 讲讲http和https

这里参考 [「查缺补漏」巩固你的HTTP知识体系](#)

HTTP

「http优点」：

- ①灵活可扩展。一个是语法上只规定了基本格式，空格分隔单词，换行分隔字段等。另外一个就是传输形式上不仅可以传输文本，还可以传输图片，视频等任意数据
- ②请求-应答模式。通常而言，就是一方发送消息，另外一方要 接受消息，或者是做出相应等
- ③可靠传输。HTTP是基于TCP/IP，因此把这一特性继承了下来。
- ④无状态。这个分场景回答即可

「http缺点」：

- ①无状态，有时候，需要保存信息，比如像购物系统，需要保留下顾客信息等等，另外一方面，有时候，无状态也会减少网络开销，比如类似直播行业这样子等，这个还是分场景来

说

- ②明文传输，即协议里的报文(主要指的是头部)不使用二进制数据，而是文本形式。这让HTTP的报文信息暴露给了外界，给攻击者带来了便利
- ③队头阻塞，当http开启长连接时，共用一个TCP连接，当某个请求时间过长时，其他的请求只能处于阻塞状态，这就是队头阻塞问题。

HTTPS

HTTPS 要比 HTTP 多了 secure 安全性这个概念，实际上，HTTPS 并不是一个新的应用层协议，它其实就是 HTTP + TLS/SSL 协议组合而成，而安全性的保证正是 SSL/TLS 所做的工作。

区别：

- HTTP 是明文传输协议，HTTPS 协议是由 SSL+HTTP 协议构建的可进行加密传输、身份认证的网络协议，比 HTTP 协议安全
- HTTPS比HTTP更加安全，对搜索引擎更友好，利于SEO,谷歌、百度优先索引HTTPS网页
- HTTPS标准端口443，HTTP标准端口80
- HTTPS需要用到SSL证书，而HTTP不用

写在文末

这是第一篇有关我整理面经中题目的文章，之后还会继续更新，如果你觉得我写得还不错的话，可以给我点个赞哦^^。看了一些面经以及面试题之后，还是觉得自己要更加努力才行，因为还有很多没完全掌握的知识点，如果面试官稍微问深一点，可能我就会答不上来了，keep going!