

2022年上海前端面经分享【携程、字节跳动】

前言

背景：一年半经验前端，今年三月初提了离职，想着趁着金三银四期间寻觅好的工作机会。不幸的是，三月初全国各地包括上海爆发疫情，直接导致许多公司开始缩招和锁 HC，金三银四变成了铜三铁四。四月份上海疫情越发严重，最终导致全城封控，只能在家准备线上面试。

（寄！）

疫情开始的阶段，整个社会的氛围都很消极。没有办法，只能边投简历，边准备面试题，边调整心情。疫情封控期间只能在家闲着，所以利用这段时间，巩固了一下 JS 基础，在 Leetcode 上刷了一些算法题。万幸的是，在这期间收到了携程和字节跳动的面试邀请（感谢万分）。那么废话不多说了，以下是具体的面试过程：

携程（旅游研发部）

技术一面

1. 常见的 React Hooks 有哪些？
 1. useMemo 是怎么实现性能优化的？
 2. useRef 的应用场景？
2. 该怎么实现【一套代码，多端运行】，说出你的想法？
3. Taro 实现跨端的底层机制？
4. 能说说你会怎样进行前端性能优化吗？
5. Chrome Devtools 的 Lighthouse 中的 LCP 是什么意思？该怎么减少 LCP 时间？

技术二面

1. 前端性能优化方案？
2. React.memo 和 shouldComponentUpdate 的作用？
3. ReactDOM.render 的流程？
4. 算法题：实现数组去重（要求最佳时间复杂度）

```
function unique(arr) {  
  const map = new Map();  
  for (let i = arr.length - 1; i >= 0; i--) {  
    if (map.has(arr[i])) {  
      arr.splice(i, 1);  
    } else {  
      map.set(arr[i], true);  
    }  
  }  
  return arr;  
}
```

5. 算法题：实现获取数组第二大的数（要求时间复杂度 $O(n)$ ）

```
function getSecond(arr) {  
  if (arr.length < 2) return null;  
  let max = arr[0];  
  let second = arr[0];  
  for (let i = 0; i < arr.length; i++) {  
    if (arr[i] > max) {  
      second = max;  
      max = arr[i];  
    } else if (arr[i] > second) {  
      second = arr[i];  
    }  
  }  
  return second;  
}
```

业务三面

1. 介绍项目及主要的使用场景。
2. 做过哪些前端性能优化，你认为项目中最主要的性能瓶颈是什么？
3. 你做的项目的推进流程是怎么样的？你是怎样进行项目的时间管理和规划？
4. 你是如何跟项目经理、后端等同事对接项目的？

字节跳动（抖音电商）

技术一面

1. 项目介绍
2. 你实现了哪些自定义 Hooks?
 1. 实现自定义 Hooks: `useLocalStorage`
3. TypeScript 泛型中 `extends` 关键字的作用
4. CSS `position` 属性有哪些值?
 1. `relative` 和 `absolute` 相对于谁而言的?
 2. 怎么实现 Header 固定在网页顶端?
 3. 使用 `fixed` 造成内容塌陷怎么办?
5. 实现一个拥有以下功能的 `request` 函数:
 1. 功能一: 支持缓存
 2. 功能二: 支持异步 (返回 `Promise`)
 3. 功能三: 支持并发请求

技术二面

1. 有一个列表, 然后为列表的每一项添加一个响应事件, 你会怎么做?
 1. 描述一下 DOM 事件流
 2. `e.target` 是指向哪一个元素?
 3. 怎么阻止事件冒泡, 怎么阻止事件捕获?
 4. React 事件机制
2. 聊一聊用户登录流程
 1. `token` 和 `cookie + session` 有什么区别?
 2. 聊一聊扫描二维码, 并通过微信登录的流程
3. 聊一聊虚拟列表实现
 1. 虚拟列表的列表项可以不固定高度吗?
4. 手写 `useMyState`

js 复制代码

```
const useMyState = (initial) => {  
  const [state, setState] = useState(initial);  
  const callbackRef = useRef(null);  
  const _setState = (_state, callback) => {  
    setState(_state);  
    callbackRef.current = callback;  
  };  
  useEffect(() => {  
    callbackRef.current(state);  
  }, [state]);  
  return [state, _setState];  
};
```

5. this 指向问题

js 复制代码

```
const obj = {
  a: 10,
  add(x) {
    return this.a + x;
  },
  reduce: (x) => this.a - x,
};

console.log(obj.add(2));
console.log(obj.reduce(2));
```

6. 实现 `sum(1)(2,3)(4,5,6)...`

js 复制代码

```
const sum = (...args) => {
  const result = args.reduce((pre, cur) => pre + cur);
  return (...args) => {
    if (args.length === 0) return result;
    return sum(result, ...args);
  };
};

console.log(sum(1)(2, 3)(4, 5, 6)());
```

技术三面

1. 选一个你认为复杂度比较高的项目，描述一下它的难点。
2. 在 TCP 建立连接后，HTTP 传输数据前，这之间发生了什么？
3. 我看你文章有些关于 TypeScript 和集合论之间的关系，简单讲述一下。
4. 聊一聊 React Fiber，如果让你实现 Scheduler 你会怎么实现？
5. 算法题：买股票的最佳时机 ([Leetcode 121](#))
6. 算法题：买股票的最佳时机 II ([Leetcode 122](#))

总结

几轮面试下来最大的感触是，面试官会针对你的项目去延伸，进而广泛地考察你的基础知识。携程更加注重性能优化方面的考察，在通过三面过后还会有一个综合测评和英语测评（综合测评考察基本逻辑不是很难，英语测评过了六级可以免试）。字节更加侧重于一些原理和算法的

考察，不过字节的算法题并没有想想中那么难，Leetcode 前 200 简单和中单难度的题准备一下问题应该不大。

最终顺利的拿到了携程的 offer，字节则是倒在三面。字节面试官的反馈说是：【选一个你认为复杂度比较高的项目，描述一下它的难点】这个问题答得太简单了，需要再加深一下技术深度，挖掘一下项目难点。以上就是此次面经分享的全部内容了，希望对你有所帮助。