**求职意向：前端开发工程师**

reasum_line

**个人基本信息**

姓 名：赵元达 工作年限：2年

性 别：男 学 历: 本科

年 龄：24 专 业：计算机科学与技术

联系方式：15731779921 毕业院校：河北科技大学

E-mail ：yuandabest@163.com 毕业时间：2021年 6 月

政治面貌：中共党员

reasum_line

**专业技能**

基础：HTML5,CSS3,JavaScript,Typescript，Vue(2/3)系列周边工具包，浏览器缓存（localstorage）

业务：独立实现后端管理系统、H5页面（App混编）、小程序、公众号开发，也写过公司内部组件库（不过是基于vue3 naviUI）

工具链的编写：自己书写了一套工具包能够开箱即用以上的开发模板

优雅的编码风格：严格要求自己代码复用性，以及函数单一执行原则。

reasum_line

**项目经历**

**项目一：玩吧APP的页面开发**

**项目时间**：2020.10-2021.7 **公司：北京默契破冰科技有限公司**

**软件架构**：

框架：Vue.js、组件库：公司内部组件发布在npm私包上、架构支撑：VueRouter、Vuex、axios等工具支撑：Vue-CLI、webpack、ESLint、Git版本工具以及commitizen规范工具。

**项目经验：**

1. 协同后端、测试、产品、UI一起开会研讨功能需求，并与项管沟通排期；

2、根据产品以及UI出的图进行静态页面的开发，再此过程中压缩图片；

3、通过后端给出的API接口获取数据，将数据添加进模块中然后进行测试并联调；

4、协同后端、测试、产品、UI再进行验收并上线。

**技术描述：**

1、使用Vue.js的前端框架，使用VueCLI的脚手架工具快速创建项目，进行项目的开发；

2、基于Webpack工程化开发解决方案；

3、基于 PostCSS 的移动端 REM 适配解决方案（公司内部框架自带适配）；

4、使用Element-ui的前端UI组件库，快速构建页面，提高开发效率；

5、基于Vue Router的路由管理方案和Vuex的状态共享方案，大部分走的是公司内部路由（特定的方法表示路由）；

6、基于axios的请求库，向服务器发请求，来获取和操作数据；

7、基于JWT的Token和本地存储localStorage来实现状态维持解决方案（公司封装了）；

reasum_line

**项目二：收费系统**

**项目时间**：2021.8-至今 **公司：华安科技技术有限公司**

**软件架构**：PC端和微信公众号（h5页面），技术vue系列 Ant-DesignUI库、Vant移动组件库。

**项目经验：**

1、PC端前期路由的划分，因为这个系统内部小功能比较多，所以单独拆成组件方便管理，其次数据通信采用了vuex结合localstoreage；

2、优化方面实现了部分图片懒加载，以及虚拟列表，然后代码分割结构清晰；

3、利用evenloop原理解决一些逻辑上无法避免的页面渲染问题，例如nextick；

4、大部分采用了回调函数做封装尽量做到了同步编程，减少因为同步异步问题造成的bug；

5、微信公众号使用各种微信提供的js库（支付库）因为是在微信内置浏览器上进行操作的，然后鉴权做了封装；

6、学会看微信开发者文档，并且封装了一套开箱即用的H5基础搭建模板，包含获取code等等，基本思想都是登录第三方鉴权；

**项目三：智慧平台**

**项目时间**：2021.8-至今 **公司：华安科技技术有限公司**

**软件架构**：PC端和App（h5页面），小程序（以前的项目）、技术vue系列 Ant-DesignUI库、Vant移动组件库。

**项目经验：**

1、PC端采用了Vue3和Typescript进行的开发，打包工具采用了Vite，整体进行了技术升级；

2、功能上添加了一些二次封装的组件，比如form和table组件还有一些echarts组件；

3、对接百度地图，然后做一些数据之间的通信，以前做的是iframe嵌套的地图（定制化）需要使用postMessage进行页面通信，后来直接采用了Vue版的百度地图更方便了原理都一样；

4、小程序部分做的是以前项目的bug修改，我并不是熟悉小程序的原理，但是会根据文档修改，我感觉这种依附于其他平台开发的只需要关心怎么用即可；

5、App其实还是做的混合开发也就是写h5页面，里面比较有价值的实现了离线缓存的功能，采用indexDB，里面书写了大量的回调函数但是保证了同步性和异步性，并没有对业务逻辑造成隐患Bug；

**技术产出**

1、针对PC、H5、库函数都做了一个基础模板；

2、书写了canvas的事件系统和渲染系统；

3、阅读了Vue2/Vue3的源码，粗略的观看了vue-router和vuex的源码（个人感觉不重要），真正的使用到了vue3的monorepo风格以及里面所用到的缓存机制结合EventLoop（nexttick原理）这个在canvas的事件系统里面的异步渲染中有使用到。

4、学习并参加开源项目工具链（pi）了解了node可以做很多事情，也理解了工具链开发的基本套路

5、坚持重构自己知识图谱

6、不足之处：单元测试写的太少了，未来可能严格要求自己使用了。Webpack的源码理解还是不够深，以及babel的源码设计层面，只停留在简单的理论层面，并不能有很好的输出。

**知识产出：**

Canvas渲染器、事件处理器：<https://github.com/zhaoyuanmeng/ycanvas>

基于上面ycanvas的hooks函数包：<https://github.com/zhaoyuanmeng/ycanvas>

前端监控：https://github.com/zhaoyuanmeng/monitor

个人博客：https://[zhaoyuanmeng.github.io/mydocs/#/](https://zhaoyuanmeng.github.io/mydocs/" \l "/" \o "https://zhaoyuanmeng.github.io/mydocs/#/" \t "https://github.com/zhaoyuanmeng/_blank)