**联系方式**

1. 手机：18500121537
2. Email：erhaier\_hai@126.com

**个人信息**

* 张易/男/1988
* 学历：本科（结业）
* 工作年限7.5年
* 期望职位：web前端开发
* 期望薪资：26K
* 期望地点：杭州

**工作经历**

**公司名称：软通(外包) 2020年9月-至今**

项目名称：linke/dayuntu,linke/aci,linke-component（蚂蚁金服）

项目感想：

大运图是一个类似于百度搜索引擎的一个项目，但搜索的内容都是基于阿里内部一个羽雀交流平台；项目也只是对内不对外；

linke-aci是一个管理工程流水线的平台，每个工程可以关联一个仓库，仓库的yml文件定义了一个流水线的执行过程，yml文件感觉就像仓库的检验员和推销员；检查这个仓库的代码是否规范，检查项目的测试覆盖率等；

Linke-component是一个npm包，主要负责包的维护与发布，提供了一个流水线的UI实现；

通过这些项目感觉搜获满满；特别是代码写完还有阿里的大佬帮你审核，指出你代码的不足，真的是太感谢了！阿里的人给人感觉都在追求完美，比如一个名字起的不妥当，非最优解都会给你打回来，让你慢慢改；应用的技术：react hooks typescript mock

1 typescript：初次体验typescript感觉太棒了，typescript极大幅度的减少了代码出错的可能性，同时也提升了开发效率，规范了数据格式；也让一些JavaScript原生方法更易使用；让一些自定义组件更易用更规范；

2 mock数据： 以前都是和后台联合开发项目，项目的运行得依赖后台制造的假数据，有了mock数据之后，使得前端可以自定义数据，使得前端一些测试变得更容易，开发更高效，也使得项目更稳健；比如：一个列表接口，列表为空时前端怎么展示空组件，列表有数据时前端如何展示列表数据，列表数据更多时的性能测试；列表接口加载时，这时由于列表数据不确定又应该如何展示？

3 oneapi：oneapi会在前端项目里自动生成后台定义的所有接口，就像前端项目里手动写的那个api文件；并且配合typescript连接口返回的数据格式，接口请求方式，接口传参格式都有定义；这进一步提升了前端项目的稳定性和开发效率；

4 hooks：曾经写过一些react项目，但从未尝试过hooks；hooks给人感觉极为灵活，使得一些业务逻辑容易封装复用；配上阿里自己研发的ahooks，antd；感觉项目真的就是在玩乐高积木，你很少需要去设置一些积木（组件，hooks），这使得项目开发很快捷，也很稳定；要说hooks有什么缺点，就是引用变量并不如class组件方便，而且还得添加对应的依赖；

5 关于如何编写可维护项目的思考：毫无疑问，项目的可维护性犹如项目的心脏，方便项目后期走向稳建，升级，功能拓展；总结如下（仅供参考）：

1语义化：语义化体现在很多方面，常见的就是命名和标签使用；好的代码命名太重要了，比如这样一段代码：return 172800000；我觉得能在第一眼看出这是描述两天的毫秒时长的都是数学高手；但如果写成这样子：return 2\*24\*60\*60\*1000；是不是更好一点，如果能这么写const twoDays=2\*24\*60\*60\*1000；return twoDays；感觉就完美了；在一个无处不在的div，我发现一些老前端都喜欢用div....顿时html5失宠了，尤其在单页应用里；我觉得写html就像写作文一样，需要用artical，section，aside，header，footer，p，small，h标签系列，ul,ol,li,summary，figure等来指明页面的各个部分都代表什么；

2 API设计：曾今有人指出jQuery犯了一个错误：不尊重原生API；比如：html vs innerhtml,text vs textContent;就连火爆的vue也是如此：v-if vs if，v-for vs for，还有vue特有的绑定事件语法，v-model，根元素设计而不是一个fragment；我觉得这就是对原生API的不尊重；用久了，会让人忘掉原生；而这正是当时用久了jquery让人非常不爽的地方；同样react中事件处理器this的指向，事件的冒泡机制也与原生不符；但总体给人感觉react的学习成本还是比较小的，更接近原生；最恶心的还是莫过于云里雾里的非主流API设计，相信有人一定根据自己的项目去封装过axios；比如封装一些http状态码的处理逻辑，后台数据格式的统一处理逻辑，最后产生一个myAxios；最关建的是myAxios的api应该与axios保持一致；尽量减少新的api，这会给后续的项目维护减少不必要的学习成本，也更容易推广；

3 减少嵌套：任何过深的嵌套都会给项目维护带来困难；最常见的嵌套就是if-else语句嵌套，html标签嵌套太深，项目结构目录嵌套太深；

if-else语句只需再嵌套一层if-else代码逻辑就会让我头大，抓狂的把多余的嵌套移除掉；比如：if(a){if(b)run()};改为：if（a&&b）run();这就会让代码舒适很多；

dom很昂贵，大家都知道，但是有时因为一些布局的需要，很容易让我们去创建多余的html标签；而这个时候很可能是一个浮动就能解决的问题；比如一个经典的左 中 右布局；只需左浮动，右浮动，最后那个元素来个margin就可以达成这种布局；相比于<Row><Col/><Col/><Col/></Row>；会节省一些html标签；

项目目录嵌套太深，也会让人头大，比如：src/pages/home/header/nav/link 这样一个目录，为了点开link这个目录，操作了好多次；改成：

1 src/pages/home/header

2 src/pages/home/nav,

3 src/pages/home/link,

这样就会减少很多嵌套，使得项目页面目录结构很清晰；

4 单元测试：一直在写业务类型的应用，一天忙到晚，一直知道单元测试能使项目变得稳健，但面对复杂多变的业务逻辑，又应该如何编写单元测试？此时感觉到设计模式中一句话颇有道理：以一定粒度去划分你的组件；抽象出不变的模式，为这些模式编写单元测试；但如何划分这个粒度因人而定，我见过阿里内置的一些很庞大的组件，我觉得封装的功能太多（比如ahooks的useRequest），会使得组件不够灵活，也会让人困惑；

5 心态：最初学习前端我更追逐于技术面，学习更多的API，但API你是根本学不完的；到现在我觉得作为一个程序员最重要的还是对代码的热情，对完美的不断追求；比如很多人在遇到一些比较棘手的问题时，很难控制住自己去采用黑科技；黑科技带来的结果就是不好维护；比如：全局变量，eval这种黑科技；很难想象阿里的一些老项目还有这玩样/(ㄒoㄒ)/~~；此时更需要你有一颗热情，追求完美的心，来抵制这些骚操作；

**公司名称：Infosys(外包) 2019年9月-2020年7月**

项目名称：晋升降级系统（海康威视）

项目感想：一个关于海康威视的人事管理系统，进项目时，项目差不多完工一半的样子，本人可能大概只负责了本项目的1/3.通过该项目感觉最大的收获是了解了海康威视的工作流程，代码规范，一些业务问题的解决方案（也是想做这个外包的主要目的）。具体技术性上的提升谈不上。项目应用的技术vue全家桶+hk-ui(其实是element-ui的一个翻版)

项目名称：大屏系统（正泰）

项目感想：同样进项目时，项目也已开展一半。我们的工作就是做页面，对接接口。项目采取的方案，本人并不赞同，但由于是客户方面采用的没提什么异议。项目是要展示在一面墙那样的大屏幕上，所以项目采用了一个postcss-px-to-viewport的插件。以前只是听说过这种方案，但尝试过后感觉一点也不好。仔细观察下vscode,word文档,浏览器的界面他们并不会随着屏幕的尺寸变大，去变大UI元素。所以本人一向反对这种方案。最后项目完工时，客户觉得图片显示模糊，我们建议提供矢量图，或者提高大屏分辨率。项目应用的技术vue全家桶+echarts

**公司名称：微羽智能科技有限公司 2018年5月-2019年8月**

项目名称：微羽智能(app Store 搜索微羽万联即可查看该项目，现在该项目只保留了H5的统计，智能两项；其余均被原生替代）

项目感想：此APP类似于小米，欧瑞博的智家；一款通过手机软件控制硬件的应用；开发时，会不时的跟硬件工程师打交道。开始开发时都不知道什么叫混合APP开发，然后。。。开始入坑。。。整个项目基于vue vue-router mint-ui echart mqtt，ios 以及一些小插件如 vue版本的swiper 裁剪照片组件 滑动删除组件;通过mqtt协议+阿里云物联网平台做中介与硬件通信，达到控制硬件的目的；

遇到的问题：

1 如何确定设备离线，设备离线引起原因复杂比如断网，断电；最终解决方案为通过心跳机制监测设备是否离线；

2 过多的mqtt消息 在客户端处理导致的页面卡顿：方案有两种：一种是启用webworker单独处理；另外一种就是将消息处理过程异步化（不同与webwoker的是，webworker单独启用了一个线程，而该方法还是占用主UI线程，只不过当处理时间达到50ms时就延时向后）；

项目名称：微羽办公，微羽客户端，员工端，小程序版智家

项目感想： 做了好多小程序，但目前搜索的到的大概只有微羽办公这一个，小程序还是感觉社区有点小，开源插件不是太多；

**公司名称：未来付 2016年3月-2018年5月**

项目名称：码上按摩

项目感想：项目比较突然，而且头一次接触小程序，不过还好小程序的开发类似于vue,而且不用自己配置路由，相比于传统网页 小程序有自己的组件，不用考虑语义化；css 无较大差别，js方面还提供了更稳定便捷的接口；给人感觉小程序更像一个微信内置的APP 提供了比web更大的权限，比如有相关蓝牙，WIFI的接口，可以监听截屏事件，震动等；项目采用响应式布局，到现在项目依然处于开发阶段，暂没有项目地址。应用到的技术css3 javascript es6

项目名称：恋爱保险

项目感想：来公司时项目已初步成形，项目是由前微软安全架构师汤姆大叔带队制作的，好遗憾，本人来到之时汤姆大叔已走俩月有余，大神的腿没抱上，只能临摹下汤姆大叔留下的项目了，恋爱保险在技术框架上，只引用了zepto 的选择器引擎，首页的H5动画引入了swiper，在布局上和本人一样采用的也是响应式布局，至今我依然不太喜欢比较流行的rem布局，特别是在图片尺寸设置这一块，图片的本身尺寸是固定的，依手机屏幕任意放大或缩小，只会影响图片显示质量，增加图片渲染时间；项目页面在服务器渲染，并未形成前后端分离的形式，直到现在本人只是一直在维护该项目，由于项目已下线无法访问；应用到的技术 css3 javascript swiper zepto fastclick;

项目名称：金榜题名保

项目感想：该项目跟恋爱保险流程差不多，一个H5宣传页，然后就是一堆表单页，项目是我跟我一个小伙伴一起做的，用git 进行版本控制，H5页我制作的有一版，另一个小伙伴也有一版，遗憾的是最后该项目未跟中国人寿达成合作协议，黄了；涉及到的技术：html5 css3 swiper zepto fastclick;

H5动画页地址(移动端)：

<https://chanelnumberfive.github.io/china-life-guide-page/>

后续做过好多项目，但思路基本上同上面两个项目大致相同，一个H5宣传页，然后一堆表单页，以下是一些项目的H5动画地址，和基于这些项目总结出来的一些常用插件

1 金婚险的弹幕动画(PC端)：

<https://chanelnumberfive.github.io/Barrage-danmu/>

2 非吃不可介绍页(移动端)

<https://chanelnumberfive.github.io/fei-chi-bu-ke/html/%E4%BB%8B%E7%BB%8D%E9%A1%B5.html>

平时也对webgl这块做了一些研究，但是webgl这块，资料感觉太少，好的书也少，相关作品如下（临摹）

2 地球demo

<https://chanelnumberfive.github.io/practise/webgl/earth.html>

3 粒子动画

<https://chanelnumberfive.github.io/practise/webgl/particle.html>

通过学习webgl，感觉就像接触了计算机底层一样，由此也深刻体会到了数学，物理学在计算机中的应用；比如：碰撞算法，图片灰度，模糊算法；

**公司名称：中创聚智 2014年10月-2016年3月**

项目名称：中国环境投融资网

项目感想：负责项目的切图与交互（主要采用渐进增强，向后兼容的方式，兼容性IE7+），应用到的技术jQuery 原生js html5 html5shiv.js css2 css3 项目前后经历四个月时间，后期也帮忙去测试，写产品文档；

公司的主要客户是政府机关，平时主要任务就是帮忙更新，修改政府机关的相关网页及兼容问题；再有就是协助flex前端制作一些单页应用的相关页面；

**技能清单**

精通javascript html5 css3

**本人短板**

1 本人做不了全栈开发；