

前端开发工程师-黄健（4.5年前端开发经验）

个人信息

- 黄健/男
 - 本科/华南农业大学(2012/09-2016/06)
 - 工作年限：4.5年
 - 期望职位: 前端工程师
 - 期望薪资: 面议
 - 邮箱: huangjianchn@hotmail.com
 - 手机: 18124283876
 - 微信: 18124283876
-

技能清单

- 熟悉react、redux原理并能熟练使用react各种配套技术栈
- 熟悉react hooks, 用react hooks进行组件以及整个应用的开发
- 熟悉ES6、ES7、ES8(如async await, Object.values()等)
- 熟悉umi、dva、Ant Design Pro。熟悉redux-saga
- 熟悉原生JavaScript: 闭包, JavaScript异步编程, event loop, 作用域, 面向对象思想, 原型链等; 熟悉函数式编程思想
- 熟悉使用Canvas, antv/G6等前端可视化工具, 曾实现点连成线、线连成框的的画图选区功能（类似高德地图的测距功能）
- 熟悉正则表达式, 利用正则表达式和node去批量替换代码
- 使用kotlin开发安卓app, 并使用webView构建复杂的页面。
- 精通DIV+CSS布局, 精通HTML、CSS, 熟悉W3C标准, 熟悉页面架构和布局, 熟悉DIV盒模型, BFC, flex布局等
- 熟练使用lodash等工具库
- 熟练使用git、SVN版本管理工具, 使用SVN、github多人协作开发
- 熟悉websocket原理, 利用websocket协议做了两个在线对战小游戏-五子棋和俄罗斯方块对拼
- 能设计可复用的UI组件（比如: table、复杂的dropdown（比如带多选、全选、tabs、搜索框等功能）等）
- 熟练使用less, sass开发工具

工作经历

- 杭州阿里巴巴城市大脑项目组（外包）（2019年9月-至今）
- 富卫人寿保险有限公司（2018年4月-2019年9月）
- 广州威尔森信息科技有限公司（2016年4月-2018年4月）
- 广州巨腾信息科技有限公司（2015年10月-2016年4月）

项目经历

阿里云-V计划

1.Task-项目描述：城市大脑项目，以调度摄像头为基础的城市数据管理系统。管理的项目包括车辆违规停放检测，垃圾堆放检测，占道经营检测，保安脱岗检测，人脸识别等。

2.Action-项目指责：负责一机一档、算子中心、任务中心、视频检测等模块的开发和维护

- 技术重点：
 - 在一机一档模块中，使用AntV/G6数据可视化解决方案，用户可以对摄像区域手绘一个多边形闭合团进行画图选区。选区以二维数组坐标标记，用以进行比如车辆违规停放检测等。使用useRef保存g6实例；以数据驱动documents，方便开发和debug。
 - 使用react hooks进行开发，使用react hooks开发组件、整个模块；使用自定义hooks。
 - 使用react hooks开发CoolForm组件，结合antd栅格布局，使用Row Col，可以应付各种布局需求。只需要配置一个二维数组表单数据例如：[{ label, formKey, type, rule, labelCol, wrapperCol... }, ...]，就可方便的生成最终的表单组件。CoolForm使用rc-form自行处理表单数据和校验，只需在二维数组配置中传入rule函数，就可方便的处理表单校验。
 - 使用 React/umi/dva/antd 等前端前沿技术开发。
 - 团队封装了一套开源的cool-utils，包括http请求，hooks，classnames等api；同时应用由同样是团队封装的cool-cli生成，抽离了webpack的构建配置，让业务逻辑与构建工具更优雅地分离。

3.Result-项目业绩：使用react hooks开发，具有良好的开发体验与开发效率，能迅速响应用户需求，开发出优秀的产品。画图选区准确，车辆违规停放、垃圾堆放、人脸识别等检测准确率、识别率高，用户反馈体验好。

Venus保险理赔系统

1.Task-项目描述：以agile的开发方法，快速迭代，迅速响应客户需求，交付出高效快捷的保险理赔系统。先完成一套功能齐全的基础产品，再切合客户的实际需求进行定制化。

2.Action-项目指责：负责基础产品Data Capture、Quality Control、Case Management、Pending Management、Smart Circle等模块的开发。负责日本客户定制化产品的一次查定、二次查定流程的开发。负责泰国AP、DA流程的Quality Control、Manual Assessment节点的开发。开发的内容主要是UI页面的搭建，数据层models的设计，表单校验，国际化处理等。

- 技术重点：
 - 使用 React/umi/dva/antd 等前端前沿技术开发，基于Ant Design Pro重新设计精美的UI。
 - 使用node + 正则准确快速地解决了整个项目替换国际化code的问题，一次性修改了500多个js文件：
 - 首先用node循环遍历/src下的所有文件，再用正则定位到原文件下的code，拿到这个code再去比对后台给的数据源中message、id、label这三个对象中是否存在这个code，若存在，将import封装好的formatMessageApi，替换框架自带的formatMessage方法，并把属性值'id'换成'message','id','label'这三者之一。
 - 使用node + 正则解决整个项目的validation的error message提示样式的替换需求，一次性准确定位修改了200多个js文件：
 - 封装ErrorTooltip组件，利用类似上面的方法，把ErrorTooltip插入到需要修改的Form.Item组件的

label属性上，找到Ant Design原本的error message提示的样式，使其失效。最终整个项目的表单校验的错误提示信息都以error tooltip的形式展现出来。

- 用lodash抽象了一套数据处理的方法，方便统一地处理表单数据的获取和设置。
- 联动的校验需要写逻辑代码去实现（比如入院日不能晚于出院日、不能有重复的手术code），我把相似的校验规则封装成函数，遇到相似的场景就调用这些函数，减轻代码冗余，而且代码清晰易读。
- 当处理onChange就会去调取后台接口的输入框时，使用Debounce处理输入框，这样就不会每输入一次就调一次接口，而是连续输入后停下一段时间后再调取接口；利用dva/saga的takeLatest api，保证拿到并发请求中最后一个请求的结果。
- 使用redux-saga的api race，使得用户可以主动取消等待时间过长的effect。
- 使用rc-form统一处理表单，使用onFieldsChange，mapPropsToFields，getFieldDecorator进行表单的双向数据绑定和validation的处理。封装了FormInput、FormSelect、FormDatePicker等7种通用表单组件，使得代码复用性更高。

3.Result-项目业绩：前端项目使用业界良好的工程实践，产出高质量、低耦合、高复用性、易维护的代码。因为基础理赔系统产品的完成度很高，得到了泰国、日本等客户的肯定，交付的定制化系统也很好的满足了他们的需求。

个人项目 - 俄罗斯方块网络对战游戏

1.Task-项目描述：完成了五子棋游戏之后不满足，再实现了一个有趣的俄罗斯方块网络对战游戏。

2.Action-项目指责：用websocket socket.io框架 + 原生js，完成了俄罗斯方块网络对战版。

- 技术重点：
 - 用socket.io框架，实时同步两个玩家的每一步键盘操作，包括（上->旋转，左->左移，右->右移，下->下移动，空格->坠落）。
 - 用纯js完成玩家操作（移动、旋转、坠落等），计时、计分等功能。
 - 游戏设计：用二维数组模拟游戏矩阵，外层数组储存游戏矩阵的行，里层数组用0，1，2模拟状态空，方块下来途中，方块固定。同样，用另一个小的二维数组模拟方块。
 - 一次消除2行到4行，还会通过websocket实时给对手从底部增加1-3行的随机行，增加了游戏的趣味性与双人对战的刺激性。
 - 一次消除多行，会给与更多的积分。比如同时消除4行，玩家将得到100分。

3.Result-项目业绩：游戏支持多人在线游玩，两两配对，原生js运行流畅，性能非常好，闲暇之余我都会跟小伙伴来一盘。

个人项目 - 五子棋网络对战游戏

1.Task-项目描述：某个周末闲得无聊，写了一个五子棋游戏。一个下午的时间完成整个游戏的功能，再用一个晚上利用websocket协议完成了线上对战等功能。

2.Action-项目指责：用websocket socket.io框架和react完成了五子棋网络对战版，整个项目前后端都由个人独立完成。

- 技术重点：
 - 用react开发，游戏可以记录历史的每一步的棋盘状态，同时实现了时间旅行，允许玩家切换至之前的某

一步“悔棋”。

- WebSocket协议，可以建立单个 TCP 连接上进行全双工通讯。利用这个特性，实现了两人轮流下子，同步时间旅行等功能，并且其中一方退出游戏后，会重新匹配新对手。
- 利用websocket协议能更好的节省服务器资源和带宽这个特性，实现了聊天室功能。支持所有玩家同时聊天，每个玩家还可以更改自己的昵称，并且玩家进入、退出都会在聊天室里面提示。
- 利用babel对es6代码进行浏览器内实时转码，利用react组件化的特性,利用lodash cloneDeep等函数，快速开发，用一个下午就完成了整个游戏的单机版。
- 用socket.io这个强大的框架，一个晚上的时间完成了后台的开发。

3.Result-项目业绩：游戏支持多人在线游玩，两两配对，同时配有聊天室功能，玩家在游玩之余可以聊聊天。

别克调研系统项目 - 用户自定义问卷

1.Task-项目描述：参考问卷星，做一套在线设计自定义问卷，并且可以发布、保存、收集问卷、分析问卷数据的系统。

2.Situation-项目难点：因为时间紧迫，不可能重新设计一套像问卷星这般完整的系统，所以我们决定在一个开源项目上做二次开发。但开源项目的代码已经很老的，一个小功能就有几千行jsp，难以后续维护，加上客户要求现代化漂亮的UI界面。所以我决定用react + Ant Design重构。

3.Action-项目指责：独立完成对一个前后端不分离的maven+jsp项目进行前端重构，使用了react.js + redux + ant design技术，在后台代码基本保持原样的情况下重构前端代码，用新技术手段克服了对前后端不分离代码的维护困难。

- 技术重点：
 - 利用react组件化的特性，极速研发，快速上线，交付给客户满意的产品。
 - 利用webpack打包代码，打包后体积小，性能、兼容性好。
 - 用es8的新特性async await + Promise.all解决了多层异步嵌套的问题。
 - 使用github开源项目quill实现了功能丰富、可定制的富文本编辑器（例如用户可添加emoji）。
 - 利用redux顺带解决了老旧jsp项目网络利用率低，重复http请求问题。
 - 使用可控组件，并将组件状态储存在redux，最后将总state树储存到后台，使用户在别的地方登陆时可以继续编辑上一次的问卷。

4.Result-项目业绩：重构之后的项目对比旧项目性能优异，http请求数量大大减少，没有了旧项目各种奇怪的bug，更重要的是符合客户的功能和UI新需求，整个界面给人焕然一新的感觉。

上汽通用营销决策信息在线平台项目 - 售后KPI、政策解读功能、月度进度计划功能、全国价格走势、车型城市价格对比等功能模块

1.Task-项目描述：上汽通用营销决策信息在线平台是给上汽通用客户定制做的一套定制化数据分析系统，以数据驱动营销、决策。这个项目是长期的大型平台项目。

2.Action-项目指责：采用本公司自主设计的ui框架，使用react技术栈，用webpack打包，用redux做状态管理，极速开发，快速交付产品。

- 技术重点：

- 重新设计、优化一些组件，供后来的项目使用。
- 使用高阶组件，防止对公共组件的污染。
- 用本公司自主设计的ui框架，直接看源码，锻炼了阅读组件源码的能力。
- 使用lodash、js原生map、reduce、filter、every、some、includes、Object.keys等方法处理数据，使代码清爽简洁、逻辑清晰、可读性高。
- 使用redux进行状态管理，实现状态记忆、状态重置等功能，路由跳转后能保持状态不丢失。
- 用echart实现数据可视化。
- 使用html2canvas将echart保存为图片，供用户下载。
- 调用后台接口导出excel数据。

3.Result-项目业绩：能快速响应用户需求，设计的公共ui组件高度可复用，功能丰富，上传到公司的组件库上，方便其他开发人员使用。这是公司的主推产品，每年业绩达到8千万。

WAYS行业政策解读

1.Task-项目描述：WAYS行业政策解读是一套关于汽车行业各地政策的综合分析平台，包括政策搜索、行业报告、补贴政策概览、车型列表、车型详情等8个模块。

2.Action-项目指责：编写高度可复用的组件、快速完成页面开发，后期打包上线，一手包办。

- 技术重点：
 - 使用webpack + es6 + babel，保证了性能和兼容性，也可以使用es6+语言的新特性，比如使用了transform-class-properties这个babel的plugin，就能使用类的静态属性这个标准委员会还没纳入语言标准的新提案。
 - 使用eslint规范代码，以airbnb的react规范为基础，作了少量适配，保证代码质量。
 - 使用postcss，为css类名添加路径名+文件名+hash值的前缀，保证类名不重复。结合less，完美解决各个模块间类名冲突问题。
 - 利用react的单项数据流特性，做出echart图（全国-省-市联动）-下拉框选择-数据展示三者之间联动。
 - 用原生js手写轮播图，保证页面性能。

3.Result-项目业绩：打包后的系统体积小，页面响应迅速；整个系统代码逻辑清晰，代码高度组件化，有非常高的可复用性，为后来的迭代开发打了一个坚实的基础。

自我评价

喜欢前端，喜欢新鲜事物，乐于探索，积极学习，善于阅读源码。有代码洁癖，轻微强迫症，做事认真，希望尽自己所能做到最好。如有幸被贵公司录用，定当竭尽所能，发挥自己的价值，为公司创造更大的效益。