

郭奇-Web 前端工程师

关于我

GitHub: <https://github.com/guoqqqi>

姓名: 郭奇 (本科, 信息与计算科学专业)

邮箱: guoqqqi@gmail.com

电话: 15225765020

地址: 广东省深圳市宝安区

技术能力

- 熟悉 HTML5、CSS3
- 熟练使用 JavaScript (ES6)
- GitHub
- Git
- Next.js
- 熟练使用 React
- 了解 Apache APISIX 项目
- 一定的工程化经验
- Cypress (E2E 测试)
- 初步了解服务端内容

其它

- 远程协作经验
- 异步沟通能力
- Apache 顶级项目 APISIX 活跃贡献者

项目经历

Apache APISIX Dashboard

背景

GitHub: <https://github.com/apache/apisix-dashboard>

Apache APISIX Dashboard 是一个使用户能够通过前端界面尽可能的轻松操作 Apache APISIX 的控制台。

预览

<http://139.217.190.60/> 账户密码为: admin

实现过程

1. 在该项目中学习使用 Ant Design 组件库, 掌握了如何灵活使用组件库搭配自定义 CSS 修改页面样式。
2. 使用 React 和 Ant Design 参与 Dashboard 中多个组件的研发, 掌握了 React 中常用的两个 Hooks: useState()、useEffect() 用法。在开发的过程中, 遇到了使用 API 接口返回的数

据去渲染界面时，如何将数据中的 Key 与 Value 同时渲染到界面的问题。最后经过查阅资料学习了使用 `Object.entries()` 方法经 `map()` 方法处理后解决该问题。

3. 学习用 Cypress 写了 Route、Upstream、Service、Plugin 等多个页面相关的 E2E 测试。在开发过程中，学会了如何使用 etcd 和 Go 在本地构建开发环境。完成测试之后，遇到了多次在本地环境测试正常运行，在 CI 环境中测试失败的案例，通过一次一次的修复，完善测试的过程，学会了使用 `should()`、`request()`、`wait()` 等 API 来使测试更加稳定的运行在 CI 环境中。对 Cypress 测试的稳定性有了一定的理解与认识。
4. 学习使用 `querystring`。解决了在 Page List 页面选择不同的 Pagination 时，页面刷新后 Pagination 也刷新的问题。学习如何用 `querystring` 为不同的 Pagination 保存状态，避免在刷新时，Pagination 也刷新。在开发过程中，遇到了一些开发方向的错误，由于对 `querystring` 的用法没有完全理解，使得开发过程变得困难。最后在请教小伙伴之后，找到了正确的开发方向，顺利解决。
5. 通过积极的参与该项目，阅读项目结构和代码规范，学习了如何创建清晰的项目结构，如何书写规范代码，正确清晰的使用变量名称、文件名，和对多次重复利用组件的拆分等知识。

Apache APISIX Website

背景

GitHub: <https://github.com/apache/apisix-website>

云原生微服务 API 网关 Apache APISIX 官方网站。

预览

<https://apisix.apache.org/>

实现过程

1. 在该项目中，我学习了 Docusaurus 框架的使用，在不同框架的项目中，了解项目的结构，学习如何修改页面样式，以及增添页面。
2. 尝试去将 Docusaurus v1 升级到 v2，跟着官方文档的指南，在升级的过程中对 Docusaurus 框架有了一定的了解，以 `docusaurus.config.js` 文件为核心配置相关内容就可以将一个项目快速搭建起来。导航与页脚的配置，都让我更加熟悉了该框架。遇到了升级版本之后的页面也需要修改，因为不知道如何去使用相关的组件库去更新页面的代码，最后未能解决它。接着我帮助社区上另外一位志愿者 Review 代码的过程中，学习他如何使用组件库，通过比较我之前的做法，学习我没有注意到的地方，和他相互交流的过程中更加明确的知道自己还需要学习什么，使我明白做一件事时，目标应更加的明确。
3. 在社区中 Review 其他志愿者贡献的代码是一个帮助我复习，也是一个帮助我学习新想法的过程，通过阅读他们的代码互相学习讨论，帮助我拓宽了想法，让我明白了在同 GitHub 中社区小伙伴异步交流的过程也是一个让知识沉淀下来的过程，每当我遇到相同的问题或我想做类似的事情时，我就会明确的知道该如何去进行，甚至在记不太清楚的时候我会找到相关的 PR 来重新学习它。

音播数据统计页

背景

GitHub: https://github.com/juzhiyuan/app_daofm_cn

统计听友收听津津乐道播客网络的数据页面，有收听位置数据，客户端类型数据，主要城市数据，收听场景数据，听友正在收听的节目数据，收听时段数据，以及每周收听排行榜数据。

效果

实现过程

1. 在该项目中，我学习如何调用 API 接口，将返回的数据渲染到页面中。开发过程中，我遇到了 API 接口返回的数据并不是我直接想要的数据的问题，我需要将数据处理完成后再渲染到页面中，因为对 `map()` 方法的不熟练，理解的不够深刻，经过 `map()` 方法处理后，未能正确的使用处理后的数据，最后发现是 `type` 类型错误造成，因为初次为 `useState()` 传入的类型与 API 接口返回的数据类型不一致。找到了问题的原因，最后将类型修改一致，得以解决。通过这次错误，我对类型有了一定的认识，有些错误未必是逻辑上的错误，根据报错信息一点一点纠正即可。
2. 我还学习了使用 Ant Design Charts 组件库，如何将数据与 Charts 组合在界面显示出，通过学习 Charts 组件 API 的属性，掌握如何让图表更美观，更符合需求的展示。学习到 web 世界之广，在项目开发的过程中，可以通过使用其他组件库来提高效率，快速完成开发。

音播邮件模板

背景

旧数据报表邮件模板在不同的邮箱表现不一致，通过修复这些问题以及进一步优化，制作了在各邮箱表现一致的：新数据报表邮件模板。

预览

音播每日数据报表



实现过程

1. 开发邮件模板的过程中，遇到的主要问题是微信、QQ、Gmail 的兼容问题和手机端与电脑端的窗口大小问题。
- 因为邮件模板的样式只能使用内联的 CSS 样式来修改，因此我在第一次调试完成之后发现，没有使用 %、em、rem 为单位真是一个错误的举动，使用 %、em、rem 为单位可以最大程度的避免布局错乱，如果不能使用媒体查询的方法时，使用这些相对计量单位就显得非常重要，通过对元素的宽高，加上它们的间距，计算出最优的比例，这样可以确保在不同窗口下始终保持布局不会错乱。
- 因为不同的兼容性会导致部分属性无法使用，导致在不同的浏览器下，相同的代码，显示的样式是不同的，因此只能使用 CSS 同兼容的属性，搭配百分比单位，在布局不错乱的前提下，保持样式相同。
2. 根据每次调试过后的结果，对症下药，每次调试完成的时候都必须把所有的浏览器都测试一遍，因为不知道会有哪些细节造成布局的错乱。邮件模板的开发项目中，不像其他页面那样，只需注意不同手机端与电脑端的适配即可，邮件模板还需要用手机查看效果，但是是很有必要的过程，否则无法确保没有问题。通过这次开发任务，让我养成了一个良好的习惯。

播客客户端页面模拟

背景

需求：根据用户输入的播客节目的信息实时生成主要客户端的页面效果图，以供用户预览。

预览

播客客户端页面模拟

从 RSS Feed 读取播客节目信息

请输入 RSS Feed URL

提交

节目名称

请输入节目名称

封面图

请输入封面图 URL

节目介绍

请输入节目介绍

A screenshot of a mobile application interface for a podcast player. At the top, there are two tabs: "苹果播客" (Apple Podcasts) and "小宇宙" (Xiaozhou), with "小宇宙" being selected. The status bar at the very top shows the time as 00:00:04 and various system icons. Below the tabs is a search bar with the placeholder text "< Search". The main content area features a large white square placeholder for a cover image. To the right of the image is the title "世界莫名其妙物语" in white, followed by the subtitle "世界莫名其妙物语" in a smaller font. Below the title is a purple button labeled "SUBSCRIBE" and a circular menu icon with three dots. Underneath the image placeholder, there is a rating section showing "5.0 ★★★★★" and "28 Ratings". To the right of the ratings, it says "2020-2021" and "Comedy Interviews". A paragraph of text describes the podcast: "干货和笑点不能两全? 这么想可就大错特错了! 三名美少女段子手组成了相声天团, 每周一每周...". This text is truncated, and a link labeled "more" is provided. At the bottom, the word "Episodes" is displayed in a bold, sans-serif font, followed by the word "MONDAY" in all caps.

节目名称

请输入节目名称

封面图

请输入封面图 URL

节目介绍

请输入节目介绍

创作者信息

请输入创作者信息

苹果播客

小宇宙

00:00:4

小宇宙

信号

Wi-Fi

电池

分享

更多

宇宙牌电饭锅

人间铁锅炖，万物一锅出——欢迎来到「宇宙牌电饭锅」。这里是这里...

2021 已订阅

+订阅

单集目录

节目详情

人间铁锅炖，万物一锅出——欢迎来到「宇宙牌电饭锅」。这里是东东枪的私密小电台，目前不定期更新，欢迎赶紧订阅，以免那啥。嗯。：）

See [acast.com/privacy](#) for privacy and opt-out information.

实现过程

1. 已经对页面的开发流程有了一定的经验与技巧，但是在这次开发的过程中难点是相关 SVG 素材的寻找，因为一些素材并不是你需要的，可能会有多余的圆或者其他，因此我又学习了 SVG 的相关属性，研究过后发现得到自己想要的 SVG 其实并不困难，一个 SVG 可能会有多个 path 标签，删除无用的 path 标签，使用 fill 属性填充颜色，修改相应的大小，最后拿到自己想要的 SVG。
2. 这次开发过程中我掌握了二个 CSS 相关技术点：
 - 按钮有一个 focus 的伪元素，可以设置按钮被点击后的一些样式，刚开始还想用 JavaScript 来完成点击按钮样式更改的效果，意外发现可以用简单的 CSS 来完成它。
 - 文字超过三行后末尾显示“...”。CSS 样式中使用 -webkit-line-clamp 属性就可以实现。如果要将多出的内容增加一个点击显示全部时，只需用到 input 中的 :checked 伪元素设置 -webkit-line-clamp: unset 即可，无需使用 JavaScript 即可实现。

- 如果想将在末尾“...”添加一个“展开”和“收起”的按钮，则只需在内容盒子之后添加一个空的 label 标签，使用 ::after 伪元素分别设置 input:checked 与 label 的 content 属性即可。
3. 该开发过程让我对 HTML 的 input、button、form 标签有了进一步的了解。对它们了解的更多就可以在使用过程中与 CSS 伪元素、JavaScript 搭配起来更加的得心应手。

支流科技官网

背景

GitHub: <https://github.com/api7/website>

支流科技官方网站。

预览

<https://www.apiseven.com/>

实现过程

1. 在该项目中，我为官网开发了第一个页面：招聘页面，这是我第一个开发任务，页面样式全部是由 CSS 完成，通过这次开发任务，我掌握了大量的 CSS3 属性，开发过程中让我熟悉掌握了两个点：
 - 通过运用一些常用的伪元素，:hover、:active 等来完成一些动画效果，按钮移动，超链接颜色更改等。
 - 通过使用 position 来排版布局，使用一个空的 div 标签来设置背景图，可以更好的控制背景图的大小与位置。
 - 使用 3D 属性与 opacity 属性搭配伪元素，可以完成一个盒子酷炫转化的动画效果。

完成该开发的过程让我深感 CSS 的强大，许多动态效果并不是一定要通过 JavaScript 来实现的。这也让我明白了 CSS 和 JavaScript 都需要更加深入的去学习。

2. 在该项目之后维护的过程中，官网加入了一个 Chakra UI 库，我又新学习了对 Chakra UI 库组件的使用，帮助修改了首页 blog 组件的自适应样式，学会组件库相关的 API 之后，只通过修改简单的几个属性就可以轻松调整样式，让我深感学习使用组件库的效率之高。但又不可过分依赖它，因为经常复制粘贴，会破坏自己一些已经习惯的开发模式。尤其是像我这样还在探索 web 的过程中，更应该优先将自己的好习惯保持下去。
3. 之后我还为官网增添了博客页、文章页、开发者大会页、新闻报道页、关于我们页。写了多个页面，让我更加熟练页面的开发流程，以及熟悉使用经常用到的 CSS 属性。
 - JavaScript 与 CSS 搭配使用，再加上一些组件库提高效率，结合使用起来不再那么生疏。
 - 在开发过程中遇到了一个相同的问题，有关静态资源的引入。我学到应将静态资源的数据放在一个单独的 json 文件中，经过 map() 方法处理后，用少量的代码完成多次重复的渲染操作是很有必要的。项目中不应出现多次重复的代码。在项目中我们应该减少重复的代码，使得项目更加的简洁明了。

- 文件结构的合理放置也会使项目更加清晰可读，又方便自己之后的快速寻找。即便是新手也能看出每个文件放的是哪块数据，这是我学到的：一个开发者应该具备的好习惯。

工作目标

1. 下半年：继续提升个人工程化能力，做好团队项目研发工作。
2. 明年：拓宽前端技术栈，目前局限于 React 与 Next.js 的使用，希望能深入了解技术细节（React Hook 原理、Next.js 以及相关组件原理）