

最好的 6 款 React 后台管理系统模板和框架

作者: [HiJiangChuan](#)

原文链接: <https://ld246.com/article/1647828511452>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



本文首发：《[最好的 6 款 React 后台管理系统模板和框架](#)》

React admin框架繁多，在本文里我们介绍 React 下最好的 6 款后台系统，每款均严格测试后，整理你们的优缺点方便你来挑选。同时我们给出一些实用建议，帮你避免选型时不注意可能导致的埋坑。

为什么后台系统极其重要

一个成熟好用的后台管理系统对于公司运营效率有极大提升，而对程序员来说，很多时候写后台系统实都是在“帮个忙”，原因是这些后台系统多数情况下和公司的主营业务并没有太大关系。

比如说，运营会请你写个系统来帮助管理商品，人事会请你写个系统来管理公司人员等等。因此，找个称手的模板和框架可以省下大量的时间。如果需要马上搭建马上能上线的后台系统，可以直接考虑拉云(见后文)，比模板和二次开发更快更方便，同时功能丰富得多。

如何挑选 React admin后台

有的时候选择一个后台系统框架时，一个开始没注意到的小问题可能导致之后系统完全无法使用。举例子，如果你的整套系统用了 SCSS 来写界面，而这个后台框架并不支持 SCSS，那么你还额外精力把 SCSS 集成进去。因此，在下文评测中我们尽量把这些细节展示出来。

我们用以下一些标准来评判 React 下的后台系统模板。

开发者体验

首先后台系统要能用得起来，开发者(也就是你)得尽量少花功夫在这个框架上，也就是说这个框架本的 API、组件或者接口需要设计得足够好，足够能用。否则，如果需要大量蛮力适配(brute-force)才满足需要的话，还不如不用框架自己写好了。

权限管理

一个后台系统是不是具有完备的权限管理，对于绝大多数公司来说都极为重要。比如，很难想象让一公司请的外包运营成员来直接操作公司的用户数据库。而正确的做法是给他们分配合适的权限，让一分人员只能操作限定的动作。

权限管理说起来简单，但是绝大多数前端模板中是无法实现的，必须配合后端。而是不是有好的接口就决定了与后端的权限管理对接时是否足够容易。

组件丰富程度

有一些组件对于一些后台来说极为重要，比如说如果你的后台是一个员工报销系统，那么时间选择器能就非常重要，因为可能需要注明发票报销的时间。如果你的后台需要展示服务器运行时间、用户数那么报表、图表组件就很重要。

对绝大多数后台来说，[表格组件](#)也非常重要。但是不管是表格、图表还是时间选择器，都是实现比较难的前端组件。所以要找到一个框架要刚好把这三个组件都做得好，是不太容易的事情。

因此组件的丰富程度和完善度，也是挑选后台系统的重要标准。

React 后台系统的类别

React 后台系统可以粗略地分为以下几种类别

1. 仅前端模板
2. 前后端集成系统，需要数据迁移
3. 前后端集成且有方便的数据接口

对于 1，适合后台已经成型，所有后台 API、数据库连接均已经包好的情况下。

第一种是指只有前端代码的 React 后台系统。当然，后台系统显然是用来操作后端数据的，所以不接后端”的后台系统是不存在的，选用这种后台系统的话，你需要自己提供后端接口和数据库接口等。个类型的代表有 [antd](#) 和各种纯 UI 的 react 框架，比如 [Material UI](#), [Mi-UI](#) 等。

第二种是指不仅有前端代码，同时也包含了后端服务。如果需要一些新的后端接口，则需要调整和开后端接口。通常来说，这类后台系统都是打包销售，比如 SAP, Salesforce 出售的一些 SaaS 后台系统，本文里就不展开讨论了。

第三种是指，前端后端集成，但不需要数据迁移。框架提供了数据接口，你只需要把你的数据接入即使用。这个类型的代表有 [卡拉云](#)。

那么什么情况下选择哪种后台系统呢？

情况一

如果你已经有一个成型的后台系统，现在只需要把后台界面做得更方便、漂亮，或者增加一些组件的情况下，你可以应该选择方案 1 或者方案 3。

原因是你已经非常明确你需要哪些组件，工作流的流程怎样，甚至已经有后端接口可以直接接入，那二次开发一下前端框架可能就可以满足你的需求了。或者用方案 3，可以比较轻松地把数据接口接入而不用对后台接口做太多调整。

情况二

还没有后端，只有数据库。

这个情况应该是绝大多数创业公司在寻找第一个后台系统时遇到的情况，工程师资源全部集中在公司事务上，没有人有精力和时间来开发后台系统。然而后台系统又极为重要，所以开始急冲冲地找一个方案。

在这种情况下，我们推荐使用方案 3，原因是后端接口还不成型，工作流不清晰可能还需要大量迭代，这种情况下直接开始固化后端是不明智的。

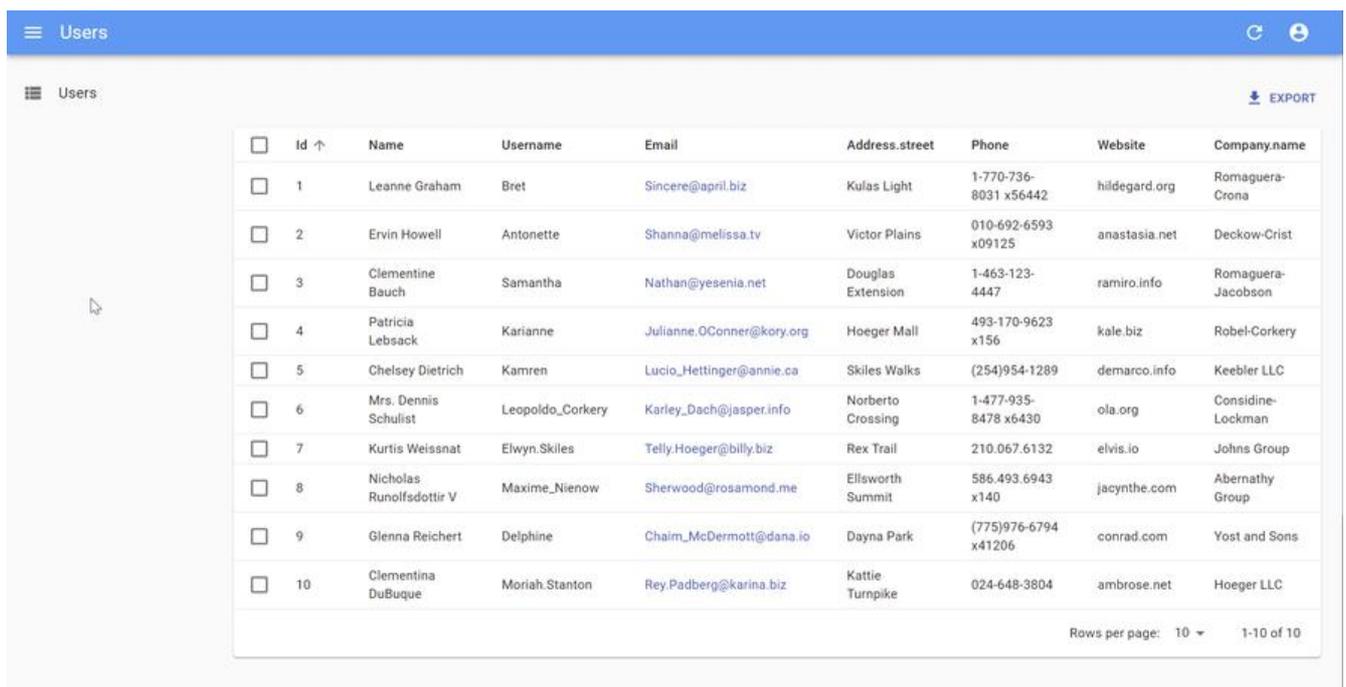
情况三

没有前端，只有后端接口。这样的情况在创业公司，甚至不少大厂中也占有一定比例。没有前端的情况下，有时候开发者甚至会直接采用直接查询和修改数据库的方式来使用“后台系统”。

在这种状态下，也推荐使用方案 3，直接将数据库接入，同时用拖拽、配置的方式生成前端。

react-admin

React-admin 是一款提供前端组件，同时还提供后端数据接口的框架。目前它在 github 上有 19.1K 星。注意，通常来说我们讲的“react后台系统”一般英文翻译为 **react admin**，而这款框架本身比较巧，名字本身就叫 **react-admin**，请注意不要混淆了概念。



The screenshot shows a web application interface for managing users. The page title is "Users". There is a navigation menu on the left with a "Users" item. The main content area displays a table of users with columns: Id, Name, Username, Email, Address.street, Phone, Website, and Company.name. The table contains 10 rows of user data. At the bottom right of the table, it says "Rows per page: 10" and "1-10 of 10".

<input type="checkbox"/>	Id ↑	Name	Username	Email	Address.street	Phone	Website	Company.name
<input type="checkbox"/>	1	Leanne Graham	Bret	Sincere@april.biz	Kulas Light	1-770-736-8031 x56442	hildegard.org	Romaguera-Crona
<input type="checkbox"/>	2	Ervin Howell	Antonette	Shanna@melissa.tv	Victor Plains	010-692-6593 x09125	anastasia.net	Deckow-Crist
<input type="checkbox"/>	3	Clementine Bauch	Samantha	Nathan@yesenia.net	Douglas Extension	1-463-123-4447	ramiro.info	Romaguera-Jacobson
<input type="checkbox"/>	4	Patricia Lebsack	Karianne	Julianne.OConner@kory.org	Hoeger Mall	493-170-9623 x156	kale.biz	Robel-Corkery
<input type="checkbox"/>	5	Chelsey Dietrich	Kamren	Lucio.Hettinger@annie.ca	Skiles Walks	(254)954-1289	demarco.info	Keebler LLC
<input type="checkbox"/>	6	Mrs. Dennis Schulist	Leopoldo_Corkery	Karley_Dach@jasper.info	Norberto Crossing	1-477-935-8478 x6430	ola.org	Considine-Lockman
<input type="checkbox"/>	7	Kurtis Weissnat	Elwyn.Skiles	Telly.Hoeger@billy.biz	Rex Trail	210.067.6132	elvis.io	Johns Group
<input type="checkbox"/>	8	Nicholas Runolfsdottir V	Maxime_Nienow	Sherwood@rosamond.me	Ellsworth Summit	586.493.6943 x140	jacynthe.com	Abernathy Group
<input type="checkbox"/>	9	Glenna Reichert	Delphine	Chaim_McDermott@dana.io	Dayna Park	(775)976-6794 x41206	conrad.com	Yost and Sons
<input type="checkbox"/>	10	Clementina DuBuque	Moriah.Stanton	Rey.Padberg@karina.biz	Kattie Turnpike	024-648-3804	ambrose.net	Hoeger LLC

react-admin 的优点非常明显

- 支持常见组件，比如 [表格](#)、文本输入框、数字输入框等
- 提供前端权限控制接口(但需要单独与后端集成)
- 提供 i18n 可以翻译界面
- 完成度较高，五年的老项目，代码质量过硬
- 代码同时支持 TypeScript 和 JavaScript，开发体验好

但它的不足也很明显。

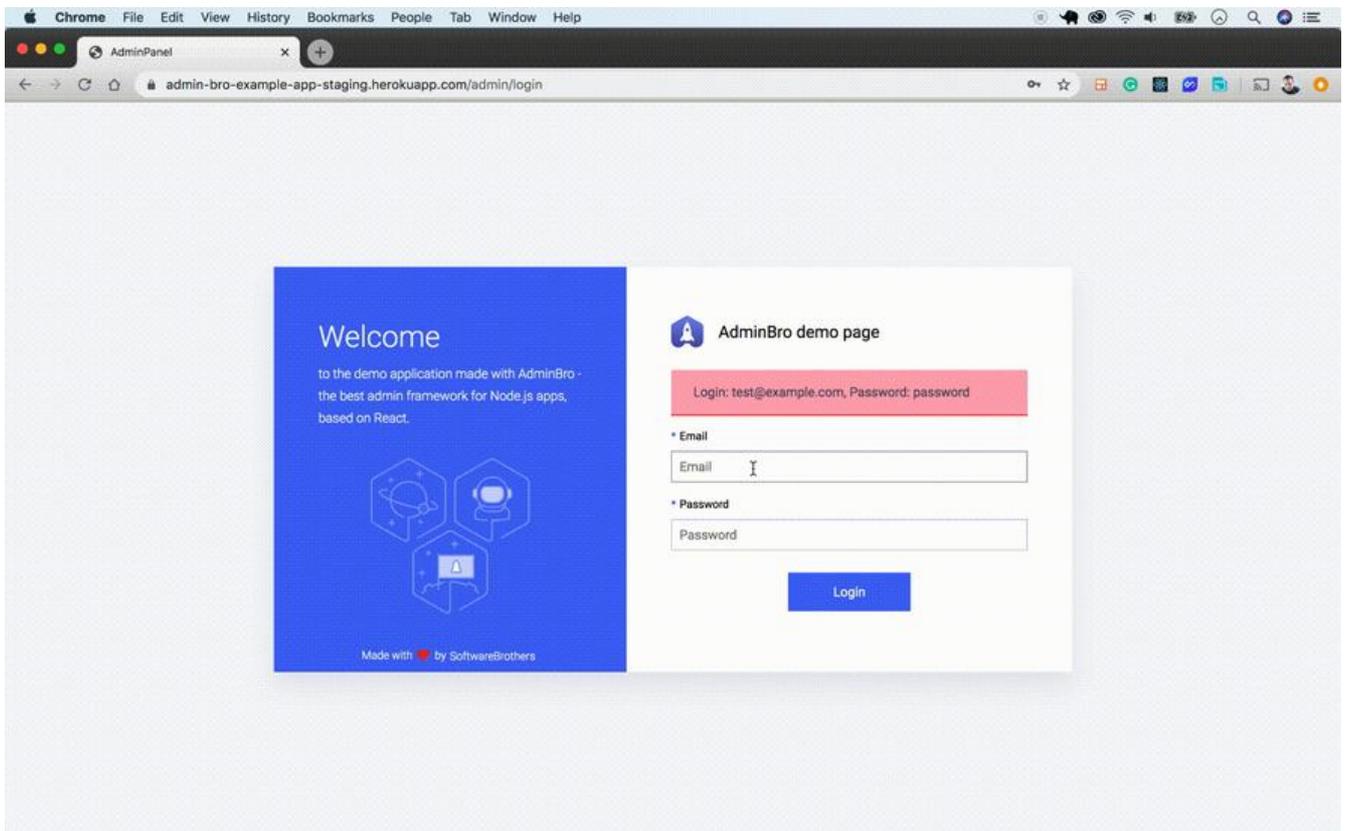
首先，组件的功能性较弱，因此需要的前端二次开发较重，只适合比较简单的使用场景。如果需要二次开发，则需要前端工程师对这个框架和 React 本身了解较深。

其次，它的数据接口层比较轻，对于不同的数据(比如REST API)需要使用不同的接口，这些都需要单集成和调试。

最后需要考虑的地方是，虽然它在 GitHub 上有大量的赞(19000)，但真正使用的项目还不如其赞多(800)，因此社区的活跃程度也并不是非常高，react-admin 的最近一次代码提交是几个月前。

AdminJS

AdminJS 与 react-admin 非常类似，也是一款主要提供前端 React 模板，同时提供轻后端接口的框。但它开始得比较晚，目前在 GitHub 上有 4800 颗星。



AdminJS 的主要优点有

- 支持部分常见组件，比如表格、文本输入框、下拉菜单
- 文档较全面，整个文档由 JSDoc 写成，非常详细
- 项目较新，维护比较勤
- 代码同时支持 TypeScript 和 JavaScript，开发体验好

但是有以下几个非常明显的缺点，我认为如果你在考虑 AdminJS 的话，不如考虑 react-admin 或卡云。

首先，AdminJS 的使用面非常窄，目前在 GitHub 上只有 188 的使用。

其次，它的代码复杂度极高，侧面反应架构设计不合理。另一方面，如果去看它的文档的话，会发现果要在 AdminJS 上做二次开发，基本上就重新学了一套新框架了，时间成本上并不划算，同时如果到问题卡住项目极容易难产。

当然即使如此，在为数不多的后台框架里，AdminJS 也是较为优秀的一款了。

卡拉云

卡拉云与其它的框架都不太一样，它不光提供一个前端框架，还提供了一个完整的前端编辑器。同时卡拉云还提供完整的后端接口，让你可以直接接入自己的数据库。在前端，你可以通过前端编辑器来置组件，包括

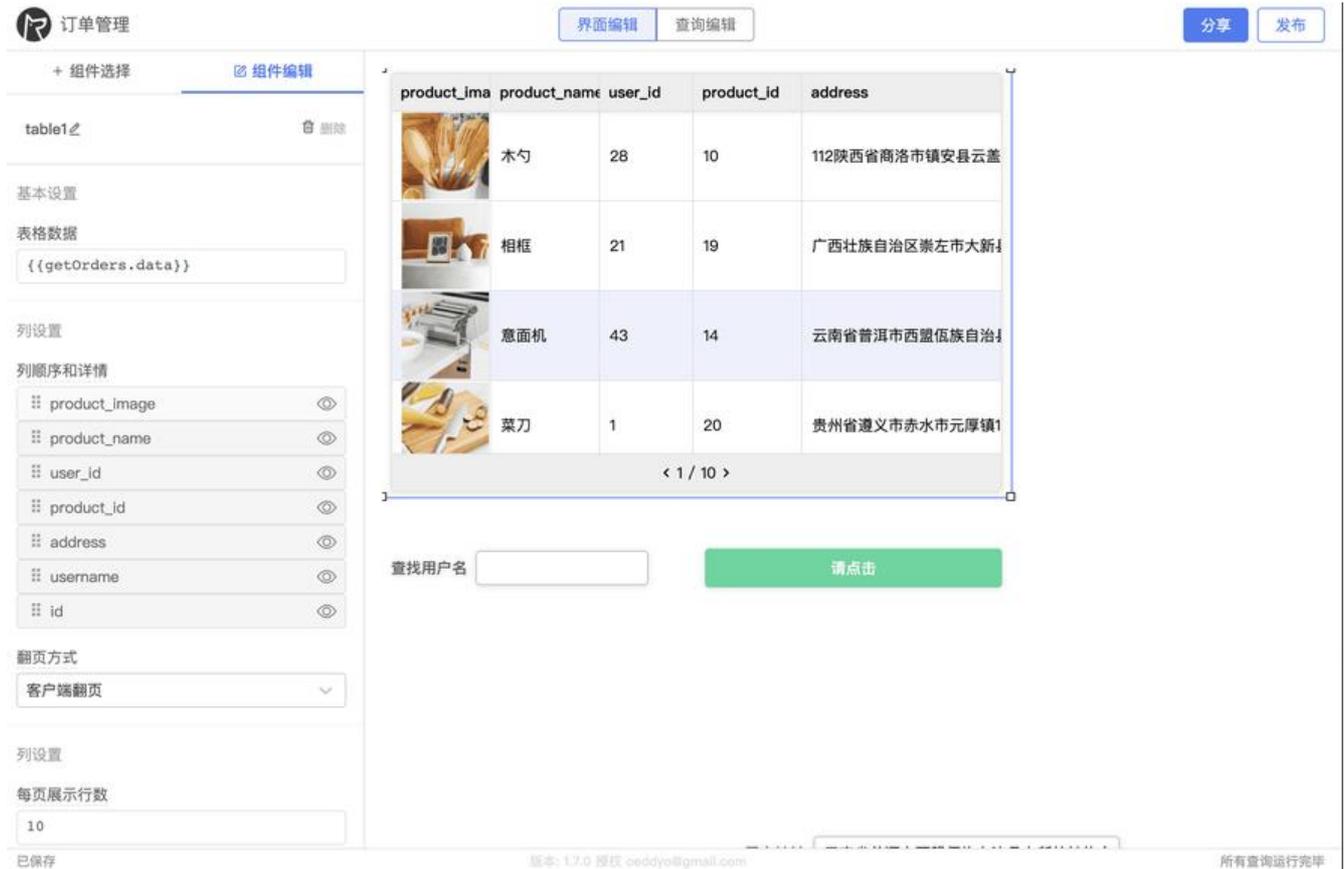
- 表格
- 图表
- 文本输入
- 数字输入
- 地图
- 时间选择组件
- 文件上传组件
- 富文本编辑组件
- JSON 编辑组件
- 数据标注组件

等等



同时每个组件都有单独的可配置项，每个可配置项均支持用 JS 来定制逻辑。因此即使是像表格这样复杂的前端组件，在卡拉云里你也可以通过细节配置，加上 JS 逻辑，来满足任意复杂的需求。比如下图为卡拉云中表格组件的配置选项，你可以针对每列动态隐藏、调整背景色，调整显示类型比如

片、链接等等丰富的配置功能。同时因为任何配置项都支持 JS，因此配置的灵活度几乎等同于直接写 eact 代码。



只要前端工程师会基本的 JavaScript(不会的话可能也不能叫前端了)，就可以在前端定制出任意复杂的后台。

在后端，卡拉云支持直接查询和更新常见数据库，如 MySQL，PostgreSQL和 MongoDB 等，同时支持 REST API 来调用你自己的 API 端口或第三方 API。

使用卡拉云的优点非常明显，你基本可以省掉要开发维护一个后台系统要操的心了，这样可以让工程专注于公司本身的业务线。同时不管是组件丰富度和完成度，以及权限系统的稳固程度，卡拉云都比面上其它框架和公司做得牢靠一些，具体详情请参考[我们的文档](#)。

antd

antd 即 Ant Design 的简写，也就是蚂蚁金服的著名前端框架。

ANT DESIGN PRO

开箱即用的标准中台前端/设计解决方案

开始使用

预览

Star 18243



antd 可以说是泛前端框架里的翘楚了，从概念上来说，它提供了很多常见的前端组件，包括表格、入框、下拉菜单等等，每个组件均提供一些参数以供配置使用。

但严格来说，antd 并不是特意为“后台系统”而存在的，因此如果你想要用 antd 来搭建后台系统的，它只是帮助你把 CSS 和组件设计这两件事情做了一些，但后台系统中需要的逻辑还是需要自己开的。比如举个例子，antd 提供了类似 [Avatar](#) 这样的组件，用来展示登录用户头像，但一个用户是登录了、登录的逻辑是怎样的，这些都是需要你自己提供逻辑的。

从经验上看，后台系统与直接面向消费者(to C)的场景不同，后台系统多数情况下可以承受较简单甚至粗糙的 UI，以换取更丰富、复杂的功能。而面向消费者的场景通常则需要将功能做得足够简单，且 UI UX 极为重要。因此，多数情况下开发后台系统时，精力均在如何把对的数据，正确地连接和展示出来，同时提供组件以供用户操作。

antd 提供了 [展示](#) 这一步，但其它的逻辑就需要前端工程师来黏合了。因此，antd 比较适合需求较固化，不经常更改，同时工程师资源有剩余的情况下。如果需要快速迭代，则应该考虑上文中提到的三种框架。

react-antd-admin

[react-antd-admin](#) 是在 antd 的基础上，由 [@jiangxy](#) 开发的后台系统。它只包含前端部分，因此如果需要任何权限控制或者与后端连接的话，需要单独做二次开发。

这个项目在早期开发频率还可以，因此在 GitHub 上积累了 1400 个星，但由于繁忙，后来就停止了发。上一次代码提交还是五年前，因此对比下来，不建议使用。同时作者自己也在 GitHub 的 readme 中也提到了安全和权限的问题：

安全/权限问题

目前对安全&权限都没考虑进去，如果有这方面的要求，只能后端校验了。在请求后端接口时校验用的身份和权限。

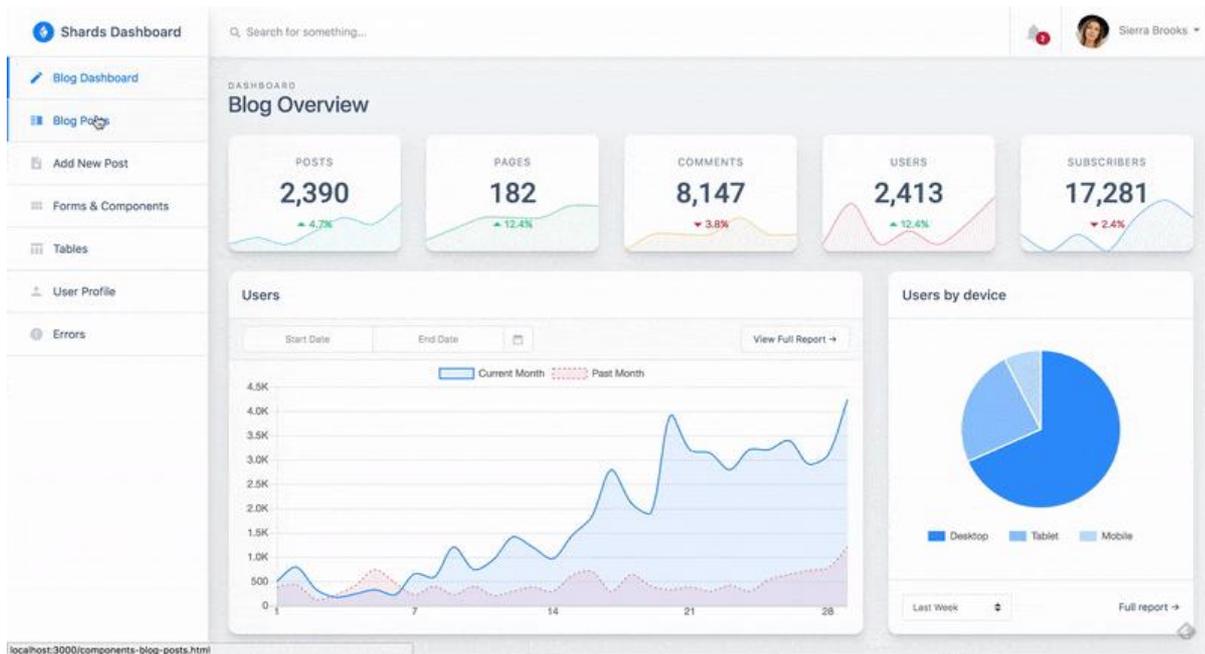
权限问题也很麻烦，感觉不太好做成通用的东西，如果有需求的话，还是定制开发比较好。

兼容性

如果你在考虑使用 react-antd-admin 的话，建议直接在 ant 的基础上自己开发，或者使用类似卡拉这样完成度较高的系统。

shards-dashboard-react

shards-dashboard-react 是一套 react 下的后台开发模板，注意这个模板的含义基本就是指它只包了常见组件和其对应的 CSS，而不包含组件具体的逻辑处理。它的界面如下



如果你的项目中，只是需要添加更多的组件，特别是 Material 风格的组件，那么使用 shards-dashb
ard-react 会比较合适。但如果连后端接口都没有，那么则需要考虑使用本文前面提到的三个框架或
统，react-admin, adminjs 或 卡拉云。

总结

本文中我们介绍了市面上最好的几款 React 后台框架，比较了它们的优缺点。相信这篇文章可以帮助
在技术选型的时候，提供一些参考。如果你对卡拉云感兴趣，欢迎联系我们。如果你对 Vue 的框架
兴趣，也欢迎阅读 [最好用的 7 款 Vue admin 后台管理框架测评](#) 和 [7 个 Laravel admin 后台管理系
推荐](#)