Create-React-App

[NextJS](https://link.juejin.cn?target=https://nextjs.org/" \o "https://nextjs.org/" \t "/Users/liujiongzhi/Documents\\x/_blank)

Remix

[Gatsby](https://link.juejin.cn/?target=https://www.gatsbyjs.com/" \o "https://www.gatsbyjs.com/" \t "/Users/liujiongzhi/Documents\\x/_blank)

UmiJS

Vite

Create-React-App是一个脚手架，它使我们能够为我们创建一个新的React应用程序。它将 webpack 和 babel 封装在一起，组成一个新的脚本工具 react-scripts 来管理整个应用，这样会减少很多复杂式，还有学习成本。

这意味着更新那些 “隐藏的” 依赖关系是一件容易的事。等待新版本的react脚本，然后进行更新。无需随意修复webpack配置的重大更改。

优势

1.较自由

可以使用任何喜欢的路由库。

2.客户端渲染

因此部署非常容易.可以将应用程序托管在任何文件主机上，如s3。客户端渲染的应用程序更易于使用和部署。

Create-React-App的缺点

1.很难定制。

如果需要自定义webpack配置，只能使用第三方工具，如craco或eject(弹出)它本身提供的功能。一旦被弹出,Create-React-App的诸多优点将消失.

2.封装导致阻止开发人员更多地了解这些重要工具是如何工作

3.对搜索引擎优化不好

由于它在客户端渲染，与搜索引擎优化作斗争，不应该用于电子商务或营销工作。

[NextJS](https://link.juejin.cn?target=https://nextjs.org/" \o "https://nextjs.org/" \t "/Users/liujiongzhi/Documents\\x/_blank)

CRA 只是一个脚手架工具，但是Nextjs是一个构建 React 应用的框架。

开箱即用，它为您提供服务器端渲染、静态站点生成、无服务器功能等等。

它是一个工具箱，为您提供创建高性能web应用程序所需的一切。

优势

1.快速。

由于服务器端渲染和静态站点生成，NextJS应用程序飞速发展。

为我们的应用进行许多性能优化，将性能作为默认设置。

2.它易于部署。

Vercel (NextJS背后的公司) 使部署全栈React应用程序变得容易。

只需点击几下，您就可以建立一个专业的部署管道。这包括预览部署和生产部署。

3.它为我们提供了API路由。

NextJS为我们提供了一种在应用程序中构建API的快速简便的方法。

如果您的应用程序使用第三方API，您通常需要自己的API来代理请求并对令牌保密，NextJS的API路由非常适合此用例。

4.易于定制。

NextJS允许我们自定义babel或webpack配置，添加webpack加载器或babel插件很容易。

缺点

1.很固执

NextJS中只有一种处理路由的方法，您不能自定义它。

NextJS仅限于其基于文件的路由器，并且动态路由仅在与NodeJS服务器一起使用时才可能。

Remix

Remix 路由基于react-router 构建, 并且你可以直接使用 React Hooks, 比如 useParams 和 useNavigate。 另一方面, Remix 内置支持嵌套布局的嵌套路由，而 Nest.js 则不支持.web 应用程序中有几种数据加载技术,Remix只支持 SSR 和 CSR.

Remix 支持开箱即用的方法操作 cookie 。您可以通过调用函数来生成 cookie，然后序列化或解析数据以存储或读取它.Remix 提供了一种内置技术，可以使用links方法动态添加样式，而 Next.js 支持 Styled-JSX 作为 CSS in JS 方案或者使用 css modules，全局样式只能写在最外层的 app.js 中

管理后台，对于数据的加载、嵌套数据或者组件的路由、并发加载优化做得很好，并且异常的处理已经可以精确到局部级别。

[Gatsby](https://link.juejin.cn/?target=https://www.gatsbyjs.com/" \o "https://www.gatsbyjs.com/" \t "/Users/liujiongzhi/Documents\\x/_blank)

Gatsby 是一个基于 React 的、免费开源的、用于搭建静态站点的框架。

，优点是有非常多的插件（plugin）和模板（starter），并且支持很多CMS（如Contentful、Neltify等），它更是一个渐进式 Web 应用生成器。通过 Gatsby 建立的网站，很容易被搜索引擎检索到，而且页面的渲染性能非常好，完美支持个人网站、博客、文档网站，甚至是电子商务网站。

### Vite

优点

1.极速的服务启动

2.轻量快速的热重载

3.优化的构建

Vite 的主要功能就是通过劫持浏览器的这些请求，并在后端进行相应的处理将项目中使用的文件通过简单的分解与整合，然后再返回给浏览器,Vite整个过程中没有对文件进行打包编译，所以其运行速度比原始的webpack开发编译速度快出许多.

缺点:

1. 生态，生态不如webpack，后者loader和plugin非常丰富,
2. prod环境的构建，目前用的Rollup，原因在于esbuild对于css和代码分割不是很友好。

### **UmiJS**

优点

可扩展，Umi 实现了完整的生命周期，并使其插件化，Umi 内部功能也全由插件完成。此外还支持插件和插件集，以满足功能和垂直域的分层需求。

开箱即用，Umi 内置了路由、构建、部署、测试等，仅需一个依赖即可上手开发。并且还提供针对 React 的集成插件集，内涵丰富的功能，可满足日常 80% 的开发需求。

企业级，经蚂蚁内部 3000+ 项目以及阿里、优酷、网易、飞猪、口碑等公司项目的验证，值得信赖。

大量自研，包含微前端、组件打包、文档工具、请求库、hooks 库、数据流等，满足日常项目的周边需求。

完备路由，同时支持配置式路由和约定式路由，同时保持功能的完备性，比如动态路由、嵌套路由、权限路由等等。

缺点

1.启动时间过长

2.热更新慢

3.框架高度封装，扩展功能不方便，自定义插件开发难度较高