

三 UmiJS

介绍

umi，中文可发音为乌米，是一个可插拔的企业级 react 应用框架。umi 以路由为基础的，支持类 [next.js 的约定式路由](#)，以及各种进阶的路由功能，并以此进行功能扩展，比如[支持路由级的按需加载](#)。然后配以完善的[插件体系](#)，覆盖从源码到构建产物的每个生命周期，支持各种功能扩展和业务需求，目前内外部加起来已有 50+ 的插件。

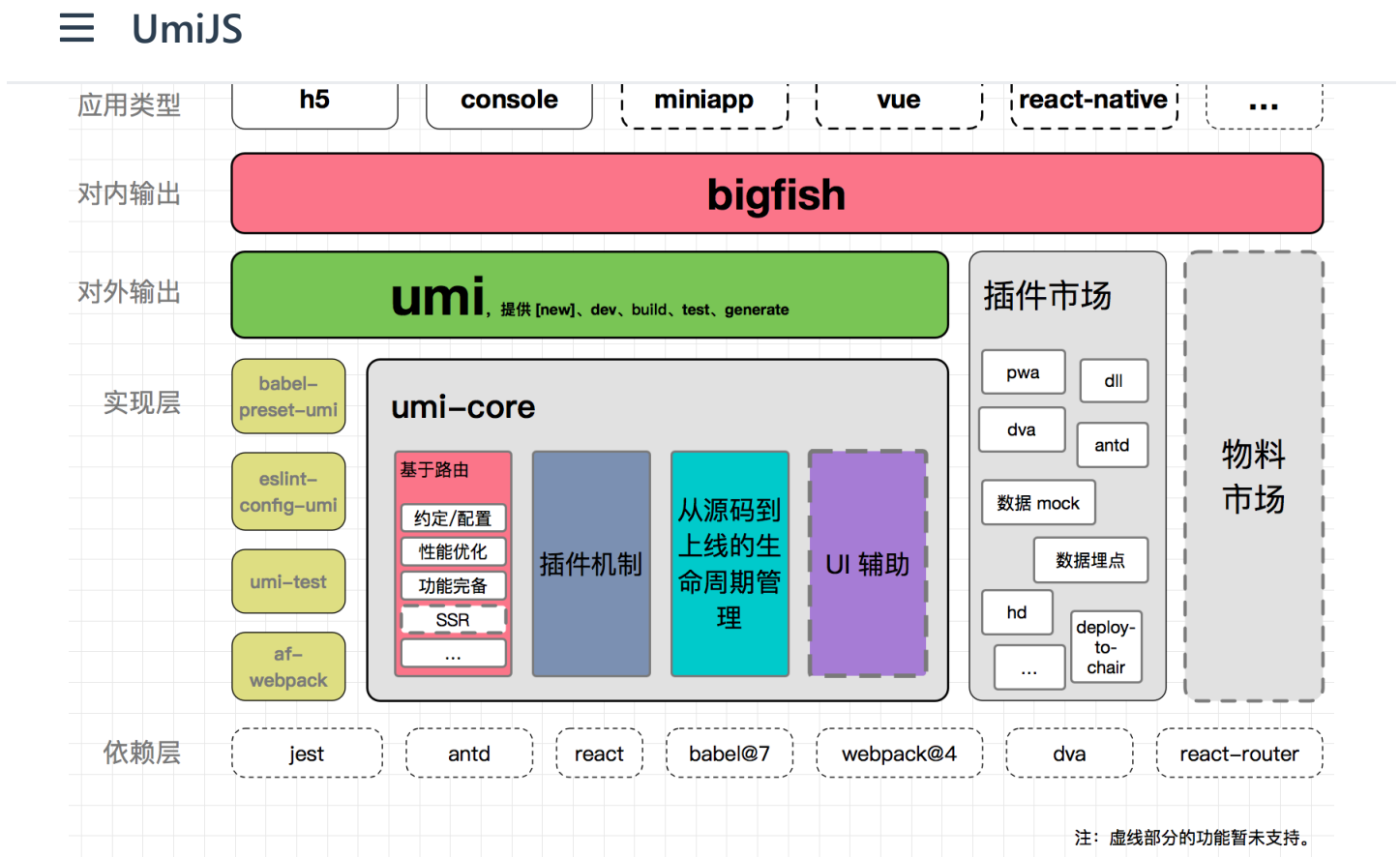
umi 是蚂蚁金服的底层前端框架，已直接或间接地服务了 600+ 应用，包括 java、node、H5 无线、离线（Hybrid）应用、纯前端 assets 应用、CMS 应用等。他已经很好地服务了我们的内部用户，同时希望他也能服务好外部用户。

特性

- 📦 **开箱即用**，内置 react、react-router 等
- 🏈 **类 next.js 且功能完备的路由约定**，同时支持配置的路由方式
- 🍷 **完善的插件体系**，覆盖从源码到构建产物的每个生命周期
- 🚀 **高性能**，通过插件支持 PWA、以路由为单元的 code splitting 等
- 🏠 **支持静态页面导出**，适配各种环境，比如中台业务、无线业务、[egg](#)、支付宝钱包、云凤蝶等
- ⚡ **开发启动快**，支持一键开启 [dll](#) 等
- 🛡️ **一键兼容到 IE9**，基于 [umi-plugin-polyfills](#)
- 🌲 **完善的 TypeScript 支持**，包括 d.ts 定义和 umi test
- 🌿 **与 [dva](#) 数据流的深入融合**，支持 duck directory、model 的自动加载、code splitting 等等

架构

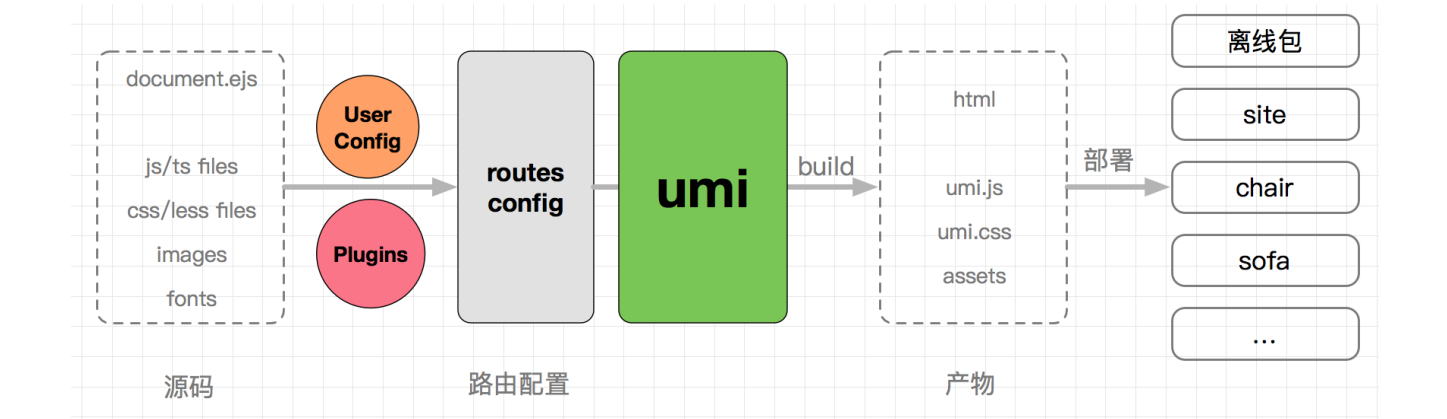
下图是 umi 的架构图。



从源码到上线的生命周期管理

市面上的框架基本都是从源码到构建产物，很少会考虑到各种发布流程，而 umi 则多走了这一步。


下图是 umi 从源码到上线的一个流程。



umi 首先会加载用户的配置和插件，然后基于配置或者目录，生成一份路由配置，再基于此路由配置，把 JS/CSS 源码和 HTML 完整地串联起来。用户配置的参数和插件会影响流程里的每个环节。

他和 dva、roadhog 是什么关系？

≡ UmiJS

- roadhog 是基于 webpack 的封装工具，目的是简化 webpack 的配置
- umi 可以简单地理解为 roadhog + 路由，思路类似 next.js/nuxt.js，辅以一套插件机制，目的是通过框架的方式简化 React 开发
- dva 目前是纯粹的数据流，和 umi 以及 roadhog 之间并没有相互的依赖关系，可以分开使用也可以一起使用，个人觉得 [umi + dva 是比较搭的](#) 

为什么不是...?

next.js

next.js 的功能相对比较简单，比如他的路由配置并不支持一些高级的用法，比如布局、嵌套路由、权限路由等等，而这些在企业级的应用中是很常见的。相比 next.js，umi 在约定式路由的功能层面会更像 nuxt.js 一些。

roadhog

roadhog 是比较纯粹的 webpack 封装工具，作为一个工具，他能做的就比较有限（限于 webpack 层）。而 umi 则等于 roadhog + 路由 + HTML 生成 + 完善的插件机制，所以能在提升开发者效率方面发挥出更大的价值。

在 [GitHub](#) 上编辑此页 

Last Updated: 2019/4/6 下午9:20:53

[快速上手](#) →