

使用 Mpvue 开发微信小程序的最佳实践

本文作者：美团点评 胡成全

前言

小程序面世一年多以来，给前端开发带来了巨大的影响，有移动应用的地方，就有小程序的踪迹。经过一年多的发展和沉淀，小程序能力不断放开，产品形态日趋成熟，已经发展成独立的端上开发技术体系，俨然与 H5、native 形成鼎足之势。但与此同时，小程序社区并不够活跃，小程序技术体系和解决方案也相对薄弱。因此，一款对开发者友好，能应对复杂应用场景，与现有前端技术契合良好的框架必将为小程序社区带来重大利好，mpvue 正是这样一款小程序开发框架。

mpvue 是什么

mpvue 是一款为解决小程序工程化、规模化开发的前端框架。她巧妙的将流行的前端框架 Vue.js 集成到小程序，使得开发者可以在小程序的开发过程中，完整的体验到 Vue.js 的能力。此外，mpvue 通过 quickstart 项目向开发者提供了完整的 Vue.js 技术方案和基础设施。

mpvue 来自哪里？

mpvue 是美团点评的开源技术项目，目前在美团点评的所有开源项目中位列第一，由专门的团队负责维护。在完成 mpvue 之前，我们经过了小程序早期的开发阵痛：无法复用现有的 web 前端代码，不能较好的支持组件化开发，无法支持 npm 外部依赖，较高的学习成本 ... 随着小程序业务的不断增多，问题规模化效应不断放大，所需的技术方案也日渐清晰：前端技术栈无缝过度，两端代码复用，提供组件化能力以适应规模化开发能力，自动化构建提升开发效率。

mpvue 发展阶段

mpvue 最终方案的形成，经历了三个阶段。

- 第一阶段：我们计划实现一个代码转换工具，可以快速的将 HTML 转换成小程序代码，以此实现前端代码复用。但工具能力定位所限，带来的效率提升只在已有代码的转化场景。
- 第二阶段：我们立足于工程化的思路，着眼于提高效率，设计了代码组件化机制。选择是简单的，基于现有技术栈，参考 Vue.js，编码实现了小程序版本的单文件组格式。我们解决了组件命名空间、数据同步等一系列问题，但无法支持复杂的应用场景，改进有限。并且对更多的 Vue.js 语法特性支持，将使得开发工作量变得异常巨大，无以为继。
- 第三阶段：计划从根本上解决第二阶段的遗留问题，实现对 Vue.js 语法的全量支持。几经探索，最后通过引入和改造 Vue.js runtime 完成。经过后续不断优化，在业务中多次实践，能力和效果最终得到验证。

mpvue 主要特点

首先 mpvue 直接支持 Vue.js 语法，对比“支持类 Vue.js 语法”有本质的区别。目前框架对 Vue.js 的语法支持程度超过90%。因此 Vue.js 框架包括的一揽子技术方案，我们都可以平移过来，作为 mpvue 的整套基础设施，这是 mpvue 最主要的能力和优势。

其次是 mpvue 的组件化开发能力。开发过小程序的同学应该有所了解，原生的公共组件提取需要拆成三个文件，在不同的入口中分别引用，使用极其不便，mpvue 将其封装的简单易行。

构建集成，小程序框架定位是一个简单的逻辑视图框架，一开始没有足够的考虑复杂场景，一旦应用的规模上来、业务变得复杂，要做到自动构建和集成需要手动完成大量工作，而 mpvue 直接提供了这套能力。

上述能力，并不能通过小程序原生框架得到，但通过使用 mpvue 都能直接获取。



mpvue 开源后续

mpvue 在内部多个项目实践后，于3月8日正式开源。美团点评技术公众号推送了文章介绍整个 mpvue 的来龙去脉。期间也获得了社区的大量关注、试用和实践，在一些技术站点或者博客都沉淀一些文章。

截止到现在，mpvue 在 GitHub 的 start 数量超过8千，issue 数量超过324个，解决259个，解决率80%。在 npm 社区，有26个关联的解决方案，其中超过20个由社区提供；此外我们组织了五个微信交流群讨论日常技术问题。

mpvue 使用场景

大家会关心，mpvue的使用场景，我们总结了一下，主要分成两类：

1. 使用小程序原生或第三方框架，到一定的阶段后使用集成mpvue做增量的开发。
2. 从零开发的时候，直接使用mpvue做新开发。

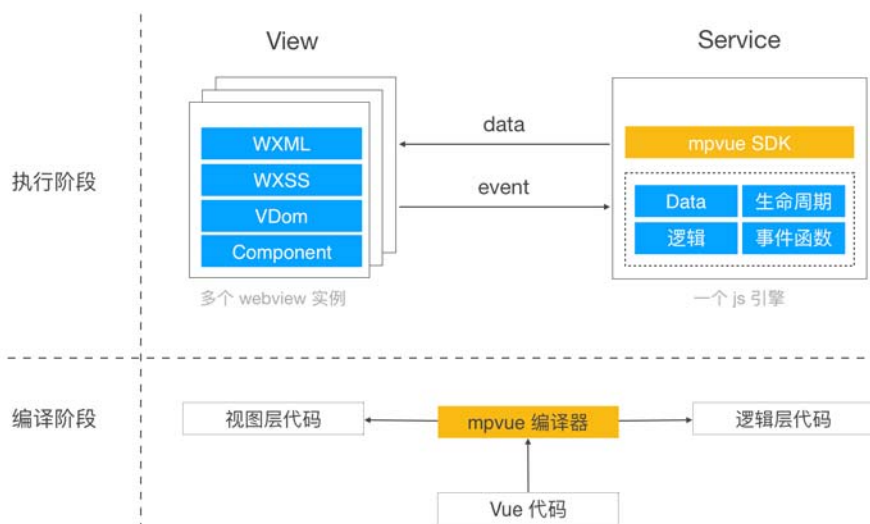
针对这些场景的话，mpvue 都提供了具体方案。从零开始使用mpvue，通过官方文档直接构建mpvue的项目环境即可。对于使用mpvue做增量开发，我们提供了 mpvuesimple 轻量化构建工具，可以对部分页面单独构建，或者将构建集成到已有的方案中。

mpvue 设计思路

小程序的定位是一个简单的逻辑视图层框架，框架将代码分为逻辑和视图两部分。视图部分代码即样式和模板，逻辑部分为js代码。在运行期小程序将所有代码做一次性加载，然后解析执行。每个页面由单独的webview 窗口负责渲染，所有逻辑代码运行在一个js引擎中。js引擎负责和素有窗口通信，并维护数据模型和响应事件。

对mpvue来说，要做的事情是让 Vue.js 语法规则的代码，在JS引擎中运行，具体的实现方式是引入mpvue的SDK。在小程序的运行环境当中，同时存在小程序对象和vue对象。mpvue 通过事件代理，生命周期打通，数据同步的方式实现了两个对象的通信。

mpvue的主要工作体现在三个方面：开发阶段，通过命令行工具快速搭建项目；代码构建阶段负责将代码编译转化成小程序的目标代码；代码执行阶段负责通过维护 Vue.js 对象来来管理小程序对象。



利用引入 Vue.js runtime 来支持的 Vue.js 语法思路的思路并不难想到，写出 demo 验证其可行性较却难了许多，但最难之处还在于具体实现。在我们看来 mpvue 的难点和核心能力主要体现在以下三方面：

1. 构建阶段的代码编译能力，即 Vue 模板视图层代码编译为小程序版本。
我们的设计是：Vue.js 维护一套完整的逻辑视图层代码；此外再单独为小程序构建出一套视图层代码用于小程序视图生成。这里的难点在于构建出的小程序模板代码需要具备一些特定能力，下文将详细展开。
2. 运行时双实例关联和生命周期互通，小程序 Page 实例和 Vue 实例需要提前在 Vue.js runtime 中建立好绑定。
具体的实现细节是：先创建 Vue 对象，在 Vue 对象的作用域中调用小程序 Page 函数初始化页面，同时将所有信息挂载到 Vue 实例，Pages 生命周期内设置 hook 调用 Vue 生命周期函数，其间所有数据从 Vue 示例上获取。
3. 另外，我们设计 mpvue 有一个指导原则，那就是：不创造规范，不引入新的学习成本。一切 Vue.js 语法我们都最大限度支持，尽力做到从 Vue.js 到小程序的零成本切换。

mpvue 最佳实践

mpvue 聚焦于解决小程序开发的效率问题，通过框架机制、语法规则来增强代码可维护性。同时也为小程序和H5的代码复用提供可操作的空间。熟悉前端开发的同学都知道，小程序和H5本身存在平台差异，部分能力无法对齐，在解决两端差异的诉求上并没有银弹。我们收到许多类似的问题：mpvue 写的代码如何在浏览器中运行。mpvue 如何解决小程序和H5的跨平台应用。针对这些问题，我们建议的方案如下：

1. 可以尽可能多的使用小程序语法特性，或者小程序组件，因为后续无需关注将其转换为H5组件。
2. 杜绝使用浏览器强相关的特性，此部分代码在小程序中无法运行。

3. 尽量少的使用不推荐的语法特性，小程序环境性能表现不佳，譬如 \$ref, filter 等。

对于 mpvue 同时兼顾小程序和H5的场景：

1. 使用平台无关语法做公共构建，此部分能力在两端表现无明显差异。
2. 平台强依赖代码做分别实现，通过抽象出适配层屏蔽平台差异，如要实现一个 slide, map, 小程序中有原生组件，但H5需要单独实现，并不能通过框架实现同等编译。思路可以类比react native。

如何快速掌握 mpvue 的能力并用于日常开发：

1. 阅读官方文档了解 mpvue 使用方式。
2. 了解小程序和H5的能力差异，便于区分出适合H5场景和小程序场景的特性。
3. 阅读mpvue项目 issue，避免常见问题，了解解决方案。
4. 对于长期建设的小程序，不断提取公共组件或解决方案，复用到更多场景。

mpvue 未来规划

mpvue 现在社区实践使用情况如何？在我司内部从去年9月份开始大规模使用 mpvue，几乎所有的场景都趟过一遍。现在正在梳理出一套从业务应用角度的最佳实践，在未来的一段时间，会把它以技术博客的方式同步给大家。

mpvue 打算做什么？框架本身的设计可以进一步优化，基础库大小有压缩空间，以适应更多的轻量化场景。社区方面，完善一套组建库方案。长远的看，我们计划把 mpvue 建设成最流行的小程序框架。



mpvue 社区建设

mpvue 本身有一份规划，我们正在征集了更多的 contributor 参与社区建设。现在在公司内外都已经有一些同学陆续加入，我们会通过一些机制来完善社区建设。

同时，大家可以看到各大公司都已经开始在做各自的小程序。对 mpvue 来说我们可以做些什么？畅想一下，未来假设我们有十个小程序的方案，如果可以通过写一套代码同时试配这些小程序...是不是很美好？

写在最后

这次分享的目的是向大家介绍 mpvue 是什么、有什么优势、能做什么、未来计划是怎样的。为大家技术选型提供参考。对于长期维护的开源项目，我们有专门的技术交流渠道，欢迎大家提供建议，或者贡献 pr，为共同促进小程序社区的发展努力。

GitChat