

01 开篇词 | 使用 Webpack 实现前端工程化

更新时间：2019-06-14 14:37:40



“人的影响短暂而微弱，书的影响则广泛而深远。”

——普希金

我是三水清，之前曾就职于新浪微博和腾讯，目前在某厂做前端架构师。这次专栏我们讲 **Webpack**，作为笔试必考、面试必问、工作必用的内容，**Webpack** 正变得越来越重要，尤其对于大型公司，会不会 **Webpack** 甚至能直接决定你是否被录用，那讲 **Webpack** 就离不开前端工程化这个大的课题。

随着多年的发展，前端越来越模块化、组件化、工程化，这是前端发展的大趋势。但无论什么“化”，其目的都是为了提升开发效率，尤其是对于大厂而言，效率直接关乎效益，所以一般大公司都会有专门的人或团队研究工程效率，我本人正是从事这样的工作，在一个知名的互联网公司负责团队的前端工程化技术建设和工具链打造。

作为一个十年的前端老司机，我经历了前端从刀工火种到现在百花齐放的各个阶段，我自己从一个互联网小白成长为能够独当一面的架构师，在整个过程中，我接触到了很多前端开发和调试技术，对前端工程化有切身的感受，自己也努力打造更加适合团队的前端工具链，希望以此来提升整个团队的研发效率。这些年，从 **Grunt**、**Gulp** 到 **FIS**，再到 **Webpack**，我带领团队做了很多工程化的尝试，目前我们团队主要在 **Webpack** 的技术上做工程化和模块化开发的探索和实践，在这个过程中我沉淀了大量的知识和最佳实践，我希望能够将这些知识分享，以让更多人受益，同时，总结和写作也是我学习输出的一种方式。

在推出这个专栏之前，我了解到很多同学对 **Webpack** 和工程化都接触不多，知道 **Webpack** 的也仅仅是把它当做一个打包工具来使用，但是随着时代的发展和你技术的成长，你会发现越来越多的问题摆在我们的面前：

- 现在开发都是模块化开发，但是模块多了，模块之间的依赖管理究竟应该怎么做呢？
- 页面复杂度提升之后，多页面、多系统、多状态怎么管理，页面间的公共部分应该如何维护，难道还是复制粘贴吗？
- 团队扩大之后，团队合作怎么做？怎么解决多人研发中的性能、代码风格等问题？
- 权衡研发效率和产品迭代的问题。

这些问题虽然不是 **Webpack** 直接可以解决的问题，但是我们完全可以围绕 **Webpack** 打造一个前端工程化解决方案来解决这些问题。很多人说会了 **Webpack** 你就变成「配置工程师」了，这句话取决于你的眼界高度。如果我们只是看到了一个 **JavaScript** 的模块化打包工具，那么我们可以满足前端开发的「温饱问题」，如果我们深入了解了 **Webpack** 内核机制，并且能够利用强大的插件系统，我们就可以实现前端开发中的「小康」。

在日常工作及准备这个专栏的过程中，我了解到很多前端工程师，哪怕是一直在使用 **Webpack** 的工程师也很少关注 **Webpack** 究竟该怎么用，都是想到哪里就上网上或官方查命令，然后闭着眼往上套，这样获取的知识零碎不系统，而且由于没有系统学习 **Webpack**，很多时候不能灵活变通，当真正出了问题时反而找不到解决方案。

其实 **Webpack** 绝不仅仅是一个打包工具，它可以做的事情太多了，系统的学习 **Webpack** 之后，我们可以基于 **Webpack** 做很多提升研发效率的事情。像 **vue-cli** 这类开发工具，都内置了 **Webpack**，每个项目的 **Webpack** 配置基本都是该项目的最佳实践，但是多个项目的时候最佳实践迁移和维护就有很多问题，所以越来越多的公司和团队会把 **Webpack** 内置化，这样做了一套构建工具的解决方案，今后有新的解决方案和优化升级，只需要升级这个工具就好了。上面所说的这些，我们在课程都会讲到。

我想说明的是，**Webpack** 是一个庞大的知识体系，不要小看了它！

学习一个新的庞大知识体系，不能够分开来零散的学习，更不能填鸭式的学习，我们应该循序渐进，从零开始直到项目实践，动手做起来才是我们程序员应该践行的学习方式。在这个专栏中，每一篇文章内都有可执行的代码可以供大家学习和实践，最后的实战部分更是可以直接应用到自己的项目中去，不仅如此，我还希望通过这些实战案例，能够启发大家在工程化实践中有新的想法，这正是我这个专栏的价值之一。

我主张学习的时候带着问题去学习，我的专栏会从零开始，一点点的带大家学习 **Webpack** 的知识体系，希望大家通过我的专栏可以解答下面疑惑：

- 什么是模块化开发？
- 为什么是 **Webpack**？
- **Webpack** 中的配置有哪些，分别可以帮我们解决什么问题？
- **Webpack** 中的概念有哪些？
- 这些概念在 **Webpack** 内核实现和原理上是怎么实现的？
- 除了打包，我们还可以使用 **Webpack** 做什么？

除了解答上面的问题，我的专栏还会就涉及到的周边知识点进行详细和深入的讲解，不仅仅能够学到 **Webpack** 的知识，还能够开阔眼界，也能够帮助大家在面试的时候解答考官问题。

另外，如果你是一个前端团队leader，或者前端面试官，在面试新人的过程中 **Webpack** 肯定是面试中必不可少的一个项目。所以我为大家分享了我积攒的这方面的“题库”，与大家一起讨论。如果你是面试官，这些题目你可以拿来直接在面试中使用；如果你是求职者，那这些题目也必然会让你在面试时有备无患，自如应对。

我希望，通过这个专栏，不管是刚入门的「菜鸟」还是想在自己团队有所做为的「老鸟」，都能够有所受益。

最后，我为大家简单介绍一下课程的具体安排：

这个专栏是从零开始入门 **Webpack**，哪怕你没接触过也没有关系，当然专栏也绝不仅仅是入门级的内容，从入门、进阶到最后通过工程化实战，带你完成系统的**Webpack**全阶段学习。我将专栏课程分为了 3 大部分（共 6 大模块），这 3 大部分为：**Webpack** 开发配置、**Webpack** 内核原理和工程化实践。

第一部分：我们要学习一个新的知识点，一定要搞明白是什么、为什么、可以做什么，所以我们专栏的第一步主要来解释 **Webpack** 是什么，以及 **Webpack** 怎么在前端开发发挥作用的，学习了这一部分算是 **Webpack** 入门了，日常开发配置都没问题了。

第二部分：学习了基础配置，我们可以称得上是个合格的「配置工程师」了，接下来我们开始深入的学习 **Webpack** 的内核实现，包括工作流程、重要的对象和实现。这部分是承上启下的部分，理解了这部分你就可以理解 **Webpack** 的流程机制，不再是一个简单的「配置工程师」了，还可以帮助团队做一些源码级别的改造和探索。

最后一步，学会了配置和原理，我们还应该发现前端研发中的现实问题和需求，能够想办法来解决问题，这才是真的做到学以致用，这部分的内容将会具有很强的实操性，有些实战可以直接在项目中使用。这一部分学完并且能够融会贯通，恭喜你，你可以借助 **Webpack** 打造自己的前端工程化解决方案了！

好了，Talk is cheap，让我们开始吧！

