

基于Vite集成的AngularJS构建前端用户界面优化路径

邱明辰

南银法巴消费金融有限公司 江苏南京

摘要: 本文聚焦于Web前端工程化技术构建用户界面技术领域,提出一种基于Vite集成AngularJS构建前端用户界面的配置方法,旨在部分前端开发人员提供技术支持。经研究表明:与其他配置方法相比,该方法解决了Vite与AngularJS集成、样式后处理、预编译、文件模块化等技术问题,有效简化AngularJS构建用户界面的复杂程度,不但能够有效满足现代前端开发需求,提高开发效率,同时亦可以显著提高用户界面在运行时的加载性能。由此可见,该配置方法具有一定的推广、应用价值。

关键词: AngularJS; Vite集成; 优化路径; 用户界面

前言

AngularJS,又被称为Angular 1.X,是一种基于JavaScript的开源前端框架,于2010年正式发布,由Misko Henvery等人创建,后由Google收购。AngularJS具有诸多特性,例如:测试驱动开发、依赖注入、语义化标签、自动化双向数据绑定技术、模块化、MVVM开发模式,为Web的发展做出了卓越的贡献。随着替代产品Angular的发布,Google公司宣布,AngularJS在2018年达到了生命周期终止的状态(EOL),将不再获得任何支持与维护^[1]。

目前,由于AngularJS不具备现代框架能力和开发范式,难以满足时代需求,通常不再被推荐使用。因此,生态缺失,市场占有率逐年减少。但目前,部分公司的管理系统、APP在构建前端界面时,依旧采用AngularJS框架技术。这是因为此类公司的存量项目承载了大量的用户流量以及日积月累的业务逻辑,难以采用新框架。因此,此类公司无法放弃对AngularJS框架的应用。

因此,为了确保业务发展具有连续性与敏捷性,如何在继续使用AngularJS构建前端界面的基础上,使得用户界面所需的现代工程化要求得到满足,是此类公司亟待解决的问题。本文将鉴于此,提出一种基于Vite集成AngularJS构建前端用户界面的配置方法,希望能够为前端技术人员提供一定的参考与启示。

1 研究背景

1.1 AngularJS技术的使用困境

Angular js是谷歌使用最为广泛的开源Angular系列

的首个版本。在2010年被公布,以其双向数据绑定和模块化开发优势而闻名,是编程行业作为稳定的框架之一,为设计者、开发者提供了大量插件与工具包,曾是构建交互式UI/UX的首选技术。但近年来,AngularJS逐渐被React、Vue等更灵活、生态也更活跃的框架所替代。

与新型框架相比,AngularJS缺陷明显。(1)Vue.js在设计及API两方面,均比Angular简单,新手可迅速掌握Vue.js的全部特征,并进行开发。(2)Vue.js更加开放灵活,可以按照开发者希望的方式,对应用程序进行组织,无需时刻遵循由Angular制定的规则。(3)在angular js中组件与指令存在不少相混的地方,而Vue.js则分的更加清晰。

由此可见,AngularJS难以适应新的技术趋势,功能上有所欠缺,性能和安全性可能无法满足现代应用的要求,市场占有率逐年减低,于2021年12月31日达到生命终止(EOL)状态,Google公司不再为Angular JS框架提供补丁与更新。

1.2 企业存量系统的现实挑战

AngularJS进入EOL状态后,由于没有补丁可用,AngularJS技术的使用,面对较大风险。但部分公司仍存在大量遗留系统。此类公司通常面临两难选择:(1)如果激进重构,利用新框架重写,风险大、成本高;(2)如保守维护,由于AngularJS生态断层,难以满足业务发展需求^[4]。因此,此类公司必须寻找AngularJS构建前端用户界面的优化路径,使之符合企业的实际需求。而Vite集成具有启动快、更新快、灵敏度高的优势,是使AngularJS框架能够继续应用的重要手段。综上,对基于Vite集成的AngularJS构建前端用户界面优化路径进行分

析,具有一定的现实意义。

2 问题与对策

2.1 问题

为了完成本配置方法必须解决如下问题:

(1) Vite支持最新的JavaScript特性和ES模块,AngularJS可能不支持这些新特性,导致代码无法正常运行。如何配置Vite对AngularJS的集成,使两者得以和谐工作。

(2) 如何合理地模块化文件分布,如何智慧地自动读取子配置。

(3) 预编译过程优化:配置在vite构建前的预编译过程,编译配置预设的静态模板,优化vite集成AngularJs的构建输入。

(4) 如何利用好vite的构建输出特性,对AngularJS中的模块和代码片段进行识别与加载。

(5) 如何通过何配置vite实现样式后处理编译过程和本地开发代理服务器等。

2.2 对策

(1) 规划工程目录结构并配置自动读取子配置策略,从而实现模块化和自动化配置管理。提升代码可维护性、复用性和团队协作效率;减少人为错误,加快部署速度,确保环境一致性,适应复杂项目需求,降低长期维护成本。

(2) 改造AngularJS框架及其它必要的插件的模块导入配置。用ES6模块配置替代AMD模块定义配置,从而适配Vite的模块系统配置要求,充分发挥Vite开发环境秒级启动、生产构建高效的优势,确保依赖解析、热更新和按需编译等功能稳定运行。

(3) 配置在Vite构建前的预编译过程。编译配置预设的静态模板,生成Vite集成AngularJs的构建输入。

(4) 根据Vite的配置规则与策略集成AngularJS。支持ES6语法,实现多浏览器兼容。确保网页在不同浏览器中功能、样式一致,覆盖更广用户;避免兼容性问题导致体验差或功能失效,提升访问量和用户满意度。

(5) 配置vite实现本地开发代理服务器、样式后处理编译过程等。

3 实施方式

(1) 规划工程目录结构并配置自动读取子配置策略。在工程目录规划设计中,本设计方法以MVC分层模式为基础,添加了“去中心化”“关注点分离”的设计模式,对配置文件进行规划,使配置管理自动化、模块化。其中,在源码目录中,每一模块均遵循mvc的分层结构,分成模型/视图/控制器的结构。而去中心化模式的落地形成“父/子配置文件”的分布,其中,父配置文件起到总调度的作用,按照目录机构文件,对目标自配置文件进行命名与自动加载。最终形成以下分布:

外层目录	父配置文件	子目录	子配置文件	备注
stroe	stroe.config			模型层总调度配置
router	router.config			视图层路由总调度配置
controller	controller.config			控制器总调度配置
modules				模型层用户界面父目录
modules		subModuleA	subModuleA.stroe	子模型配置文件配置
modules		subModuleA	subModuleA.controller	子控制器配置文件配置
modules		subModuleA	subModuleA.router	子路由配置文件配置

如表所示:在启动vite构建命令过程后,各父配置文件会遵循类似“../modules/*/*.store.js”、“../modules/*/*.controller.js”和“../modules/*/*.router.js”的规则自动匹配上分配在各个子视图目录下的配置文件,而后子配置文件中的配置项会通过编译过程汇总到父配置文件中。通过这一方式,可使开发者无需对外层目录的代码进行关注,而专心于工作涉及的子视图目录^[5]。

(2) 为适配vite的配置要求,对AngularJS框架及其它必要的插件的模块导入配置进行改造。为了使vite

可以正确处理AngularJS的依赖于模块,首先必须对AngularJS框架及其它必要插件的模块定义、模块导入配置方法、依赖注入开展源码级别的改造,使得AMD模块定义配置转变为Esbuidl模块配置,从而对Vite的模块系统的配置要求进行适配。对于非核心文件,异步加载的文件则通过在文件路径后配置参数的形式根据实际情况获取文件或文件路径。如“? raw”或“? url”。

(3) 配置在Vite构建前的预编译过程,编译配置预设的静态模板,生成Vite集成AngularJs的构建输入。在

AngularJS框架的机制中没有单独经过抽离的配置文件，其所谓的配置项是作为函数的实参。因此实际情况是配置项与代码块耦合在一起，其在框架启动阶段被执行。此外该框架路由的配置中，同步与异步的代码法具有较大差异，且配置多个路由时代码具有较多的冗余。因此，在本文所述配置方法中，便采用了预编译策略来静态编译预设的配置文件和模版文件。利用可预编译的可行性，抽象出更为精简的配置文件。此外，静态编译模版后输出文件是可再次参与vite构建的文件。即：通过预编译，最终输出的是Vite集成AngularJs的构建输入，此外也是AngularJS框架可执行的代码片段^[6]。

(4) 根据Vite的配置规则与策略集成AngularJS。待第二步改造完成后的Angularjs框架及其相关插件在vite中配置结束之后，开发人员还可在编码时写符合W3C标准的ES6语法，利用编译配置能够向下兼容。这导致开发人员进行作业时，无须对浏览器兼容问题进行考虑，有效地提升开发效率。在后续运行时，每个视图模块均配置有单独的控制器进行调度，而控制器的挂载模式则由配置文件的配置项进行控制。从而实现视图模块的独立控制器调度，提升模块间的解耦。

(5) 配置vite实现样式后处理编译过程和本地开发代理服务器。自定义Vite提供的集成化构建配置，从而实现从源代码到生产就绪包的自动化构建流程^[7]。其中，通过对生产运行环境与本地开发环境的区分，对本地开发服务器进行选择性地配置。此外，编译过程包括以下内容：样式后处理向下兼容的编译过程配置、HTML模板模块化编译过程配置；前端打包包括文件内容自动读取、合并、切割、去重、zip压缩、增加文件hash和构建可运行的前端界面的目录与部署压缩包的配置^[8]。

结束语

作为一种已经停止维护老牌前端框架，AngularJS已经难以满足开发者的实际需求，本文提出了一种基于Vite集成AngularJS构建前端用户界面的配置方法。该方法

能够使开发者在构建复杂的前端应用时，降低学习曲线，提高开发效率。得益于Vite的高效性能，最终用户体验到更为顺畅的界面操作，以此提高了用户满意度和粘性。在金融、电商、社交等行业，均可以利用这一配置方法，在短时间内构建出用户友好、响应迅速的前端应用。由此可见，该方法具有广阔的应用前景，应当做进一步的普及。

参考文献

- [1] 前端周刊.AngularJS停止长期支持；知名开源库遭开发者恶意破坏；GitHub开源趋势；Firefox 96发布.[EB/OL] (2025-07-04) [2022-01-17]<https://zhuanlan.zhihu.com/p/458327491>
- [2] 陈伟，张奥然，许信宇，等.一个基于SpringBoot和AngularJS的家庭理财系统设计与实现[J].电脑知识与技术，2024，20(12)：46-49.
- [3] 刘青丹，王舒憬，强杰.Ionic+AngularJS框架在跨平台旅游APP客户端系统中的应用[J].工业控制计算机，2018，31(01)：142-143.
- [4] 卢冬冬，吴洁，刘鹏，等.开源软件社区开发者合作网络稳定性研究——以AngularJS为例[J].复杂系统与复杂性科学，2020，17(03)：38-46.
- [5] 梁洁.基于Ionic和AngularJS框架的交互式移动学习平台设计与实现[J].智能计算机与应用，2020，10(07)：250-252+256.
- [6] 海涛，张佳毅，王智迪，等.基于Jfinal和AngularJS框架的图书馆微信服务平台的开发与研究[J].吉林师范大学学报(自然科学版)，2020，41(01)：78-83.
- [7] 海涛，张佳毅，王智迪，等.基于Jfinal和AngularJS框架的图书馆微信服务平台的开发与研究[J].吉林师范大学学报(自然科学版)，2020，41(01)：78-83.
- [8] 司桂静，崔向前，吴兴，等.基于Ionic3和AngularJS实现城域网掌维APP[J].山东通信技术，2020，40(02)：46-47.