

Web新兴前端框架与模式研究

文 / 占东明 洪家伟 陈希杨 徐礼飞 辛郢放

摘要: 随着互联网的进步, 传统的Web技术已经不能满足于人们对交互体验的更高要求, 使用React JS, bootstrap, Angular JS, H5等新兴技术成为未来前端开发的必然选择; 展示了近几年兴起的新兴技术和工程化思想, 分析其优缺点; 指出应用和应对这些技术是未来几年前端发展的重要课题。

关键词: 前端发展; 前端模块化; Angular Js; HTML5; Webpack; Backbone js; React js

引言

近年来随着前端领域的大跨步发展, 前端领域的技术方向呈现出了百花齐放的盛况。许多前端学习者在面对如此多元化的技术方向时, 不免感到疑惑和茫然。本文旨在探讨近年来前端领域里一些框架和模式的应用方向及其特点。综合考虑了客观规律和主观判断的因素, 以此为前端人员在技术决策提供科学、合理的分析。

1、前端基石与催化剂

1.1 浏览器的跃进

现在H5已经成为了一个符号, 基本上所有具有绚丽界面或者交互的Web界面, 无论是PC还是Mobile端, 都被称为基于H5。H5技术的发展以及带来的一系列前端的变革, 都离不开现代浏览器的发展与以IE为典型代表的老的浏览器的消逝。

1.2 浏览器端的魔术: AJAX

AJAX即“Asynchronous JavaScript and XML”(异步的JavaScript与XML技术), 指的是一套综合了多项技术的浏览器端网页开发技术, 可以基于JavaScript的XmlHttpRequest的用于创建交互性更强的Web应用。AJAX是一种已有技术的mashup, 多种技术组合在一起形成了其特色和优势, 早在1998年就已经开始有人使用。Google在地图和Gmail等产品中对这项技术的深入应用, 以及AJAX这个吸引眼球的名字的提出, 使其正式站在了聚光灯下, 开始吸引无数人的目光。我们知道Web应用中用户提交表单时就向Web服务器发送一个请求, 服务器接收并处理传来的表单, 并返回一个新的网页。而前后两个页面中的往往大部分HTML代码是一样的, 每次都返回整个页面内容是一种带宽资源的浪费。而AJAX应用仅向服务器发送并取回必须的数据, 并在客户端采用JavaScript处理来自服务器响应, 更新页面的局部信息。这样不仅浏览器和服务器的数据交换大大减少, 而且客户端也可以更加快速地响应用户操作。如果你用Gmail就应该知道, Gmail从来都不刷新页面, 所有的请求都是通过AJAX获取数据进行局部更新。AJAX的出现, 以及诸如EXTJS、DOJO等一些前端开发框架的出现, 也使得单页应用(Single Page Application)在这个时候流行起来。

1.3 ECMAScript

2015年是JavaScript诞生的20周年。同时又是ES6标准落地的一年。ES6是迄今为止ECMAScript标准最大的变革(如果不算上胎死腹中的ES4的话), 带来了一系列令开发者兴奋的新

特性。从目前es的进化速度来看, es后面应该会变成一个个的feature发布而不是像以前那样大版本号的方式, 所以现在官方也在推荐ES+年份这种叫法而不是ES+版本。

更让人兴奋的是, JavaScript慢慢不再局限于前端开发中, NodeJs的提出让人们感受到了利用JavaScript进行全栈开发的能力, 从此大大提高了开发的效率(至少不用多学习一门语言)。JavaScript在物联网中的应用也曾经引起一些追捧与风潮, 不过今年物联网社区更加冷静地看待着这个问题, 但是并不影响各大厂商对于JavaScript的支持, 可以参阅《JavaScript Beyond the Web in 2015》这篇文章。可以预见JavaScript在其他领域将继续大放异彩, 毕竟ECMAScript 6, 7已经是如此的优秀。

2、模块化和工程化

2.1 前端MVC: Angular/Backbone

这种模式下, 前后端的分工非常清晰, 前后端的关键协作点是Ajax接口, 规定好交互接口后, 前后端工程师就可以根据约定, 分头开工, 开发环境中通过Mock等方式进行测试, 同时在特定时间节点进行前后端集成测试。但是, 随着业务功能的愈发复杂(看看现在的Gmail), 这种模式本质上和JSP时代的Web开发并无本质区别, 只不过是复杂的业务逻辑从JSP文件转移到了JavaScript文件中而已。现在, 对于一个前端功能、交互复杂的SPA, JavaScript代码很容易膨胀(超过10万行)。很自然地, 像服务端从JSP向MVC框架转换的过程一样, 前端开发也出现了大量的MVC框架, 比较典型的包括BackboneJS, AngularJS, EmberJS, KnockoutJS。总的来说, MV*框架的提出是为了解决前端开发的复杂度, 提供一套规则组织代码、分层(MVC), 通过合理的组织和分层, 前端的代码职责明确、清晰, 便于开发与测试。

2.2 前端工程化

大部分时候我们谈论到工程化这个概念的时候, 往往指的是工具化。但是任何一个通向工程化的道路上都不可避免的会走过一段工具化的道路。前端工程化的道路, 目标就是希望能用工程化的方法规范构建和维护有效、实用和高质量的软件。

工程化的要素, 会有以下几个方面:

1) 统一的开发规范(语法/流程/工程结构)与编译工具。实际上考虑到浏览器的差异性, 本身我们在编写前端代码时, 就等于在跨了N个“平台”。在早期没有编译工具的时候, 我们需要依赖自己去判断浏览器版本(当然也可以用jQuery这样人家封装好的), 然后根据不同的版本写大量的重复代码。最简单的例子, 就是CSS的属性, 需要加不同的譬如-o-、-moz-这样的前缀。而这样开发时的判断无疑是浪费时间并且存在了大量的冗余代码。

2) 模块化/组件化开发。在一个真正的工程中, 我们往往需要进行协作开发, 之前往往是按照页面来划分, 但是会造成大量的重复代码, 并且维护起来会非常麻烦。这里的模块化/组件化开发的要素与上面的第一点都是属于开发需求。


3) 统一的组件发布与仓库。许多程序员在使用Maven前后会有很大的一个对比感,没有一个统一的中央仓库与版本管理工具,简直就是一场灾难。这样也无法促进开发者之间的沟通与交流,会造成大量的重复造轮子的现象。

2.3 Webpack

Webpack跟browserify本质上都是module bundler,差异点在于Webpack提供更强大的loader机制让其更变得更加灵活。当然,Webpack的流行自然还是离不开背后的react跟facebook。但是从现在HTTP/2标准的应用及实施进展来看,Webpack/browserify这种基于bundle的打包工具也面临着被历史车轮碾过的危机,相对的基于module loader的jspm反而更具前景。Browserify可以让你使用类似于node的require()的方式来组织浏览器的Javascript代码,通过预编译让前端Javascript可以直接使用Node NPM安装的一些库。相较于Webpack, Browserify具有更悠久的历史。

Webpack是前端开发真正意义上成为了工程级别,而不再是随意,可以看看这篇文章。笔者第一次看Webpack的时候,没看懂。当时用Gulp用的正顺手,不需要自己往HTML文件里引入大量的Script文件,还能自动帮你给CSS加前后缀,自动地帮你压缩,多好啊。不过Grunt和Gulp现在存在的问题就是需要自己去组装大量的插件,参差不齐的插件质量导致了大量时间的浪费。并且Gulp/Grunt还并不能称为一个完整的编译工具,只是

一个辅助工具。

Webpack还有很令人欣慰的一点,它支持Lazy Load Component,并且这种懒加载技术是与框架无关的。这样就避免了开发者在编码时还需要考虑固定的组件或者代码分割,毕竟在一个快速迭代的项目中还是很难在一开始就规划好全部的组件分割。同时,Webpack还支持配合了React Hot Loader的代码热插拔,可以大大地提高代码的开发效率。

参考文献:

- [1] Patrick Catanzariti, JavaScript Beyond the Web in 2015, 2015.
- [2] 徐进强, 浅谈AJAX技术在WEB程序开发中的应用, 2015.
- [3] 寸志, 前端模块化杂谈, 2015.
- [4] 阿大、慎里, 前端工程化: 云构建, 2016.
- [5] chemdemo, 基于webpack搭建前端工程, 2015.

作者简介:

占东明, 研究生, 华东交通大学, 讲师, 研究方向: 电子商务与电子政务、互联网+;

洪家伟, 本科, 华东交通大学, 学生, 研究方向: 软件工程;

陈希杨, 本科, 华东交通大学, 学生, 研究方向: 软件工程。

徐礼飞, 本科, 华东交通大学, 学生, 研究方向: 软件工程;

辛鄢放, 本科, 华东交通大学, 学生, 研究方向: 软件工程。


(上接第10页)

将电子档案的数据存储格式和逻辑结构分类,可分为结构化数据、半结构化数据和非结构化数据。在如今各个单位电子档案中所包含的数据,如DOC、CAJ、PDF等均为非结构化数据。这也是大数据时代的特点Variety,即数据种类多样。对非结构化数据管理与存储进行设计变得十分必要。高艳霞、张慎武提出采用TRIP数据库、C/S和B/S系统相结合的方法来设计了一套用于电子档案给结构化数据的存储的数据库并处于安全性考虑对用户权限进行了设计,规定每个部门分别建立自己的数据库并管理自己的电子档案。^[10]非结构化电子档案在管理上存在诸多的难点,如档案数据不兼容、档案信息冗余、档案数据之间联系性弱等。王志宇、赵淑梅将无ERMS环境下非结构化档案数据管理和ERMS环境下非结构化档案数据管理分开讨论,在无REMS环境下使用虚拟文件夹信息分类技术和桌面搜索技术,在REMS环境下在数据库中挂接非结构数据档案并使用数据库嵌入存储技术,同时要求对数据进行内容管理。^[11]关于电子档案数据的处理可以利用MapReduce技术和分布式文件系统,MapReduce是Google公司基于云计算开发的一种数据处理系统,最早用于非结构化数据的处理,现成为大数据处理的关键技术,同时Google公司开发的分布式文件系统GFS为MapReduce提供数据存储。现金除了GFS外,HDFS、FastDFS和Haystack等都是大数据时代数据存储的技术,档案管理部门应结合部门实际来选择。^[12]

2.3 电子档案的环境建设

大数据环境为电子档案的管理和利用提供了更多的可能性,是我们对电子档案的实用性产生了进一步的设想和期待,机会与风险并存,在当前环境下的电子档案工作也面临的问题需我们为其创造一个良好的环境。目前来讲我国有关电子档案尤其是在电子档案的利用与共享上存在法律不健全的问题。同时也需要我们进一步明确在实际档案管理中各个部门的权利与责任,这既是一个权责和法律的问题,也是一个技术和人员的

问题。我们应该进一步研究如何在现有技术下既保证电子档案的真实性完整性及其安全问题,又解决目前多方利益协调下的电子档案的共享问题。这急需我们在法律、道德、资源上为其创造一个良好的环境。

目前来看,电子档案管理在大数据时代如何应对数据爆炸式增长带来的存储利用以及安全问题上已经有了一定的思路,但具体操作细节以及效果如何仍然需要进一步探索和实验。为提高电子档案管理效率和效果管理人员如何做到全程关注并与其他部门积极配合完成电子自档案管理任务在大数据时代下显得更为重要,也是今后需要讨论的地方。

参考文献

- [1][4] 蔡学美. 档案工作中使用“电子档案”概念的基础和作用[J]. 档案学研究, 2011, No. 11801: 4-6.
- [2][6] 黄新荣. “电子档案”概念新解[J]. 档案学研究, 2008, No. 10102: 53-55.
- [3] 陈永生, 杨茜茜, 侯衡, 苏焕宁. 电子政务系统中的档案管理: 问题与思考[J]. 档案学研究, 2015, No. 14302: 28-37.
- [5][11] 王志宇, 赵淑梅. 非结构化电子档案数据管理探析[J]. 档案学通讯, 2014, No. 21905: 54-58.
- [7] 王萍, 张帅, 张诗敏. 国外近十年电子档案研究述评[J]. 档案学通讯, 2011, No. 20103: 58-62.
- [8] 毛海帆. 电子档案元数据方案设计与应用初探[J]. 档案学, 2010, No. 11201: 74-78.
- [9] 彭远明, 涂昊云. 电子档案安全评价指标的制定与实现方式[J]. 档案学研究, 2013, No. 13506: 65-70.
- [10] 陈永生, 杨茜茜, 侯衡, 苏焕宁. 电子政务系统中的档案管理: 查询利用[J]. 档案学研究, 2015, No. 14605: 17-28.
- [12] 高艳霞, 张慎武. 电子档案的非结构化存储方案浅议[J]. 兰台世界, 2012, No. 35202: 20-21.

作者简介: **李一**, **刘邦凡**, 单位: 燕山大学公共管理学院;
何太淑, 单位: 燕山大学档案馆。