2016年软件开发白皮书：Web开发群体5成以上为全栈开发者，全栈工程师将成为IT界新宠！

全栈开发之道

MongoDB+Express+ AngularJS+Node.js

和凌志 编著

适合想学前端技术的APP（iOS、Android）开发工程师，以及想学习后台技术的前端工程师

中国工信出版集团

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

全栈开发之道

MongoDB+Express+ AngularJS+Node.js

和凌志 编著

電子工業出版社
Publishing House of Electronics Industry
北京•BEIJING

内 容 简 介

全栈（Full Stack）是一种全新的以前端为主导的框架，框架选型聚焦在 MEAN（MongoDB、Express、AngularJS、Node.js）上。选用 MEAN 全栈技术，可以快速地实现敏捷开发，尤其是到了产品的运营阶段，其优势表现得非常明显。本书主要介绍 MEAN 全栈技术，分为入门篇、基础篇和实战篇，入门篇对全栈进行了概述，基础篇重点介绍了全栈的四个主要技术，即 MongoDB、Express、AngularJS、Node.js，实战篇则通过四个常用的实例来引导读者循序渐进地掌握全栈开发的思路。本书重在讲述全栈开发的思想，自始至终延续这样的一个主题：如何通过一种框架（MEAN 全栈），将前端和后台（端）贯穿起来，让前端工程师快速上手。

本书适合想学前端技术的 APP（iOS、Android）开发工程师，以及想学习后台技术的前端工程师阅读。

图书在版编目（CIP）数据

全栈开发之道：MongoDB+Express+AngularJS+Node.js / 和凌志编著. — 北京：电子工业出版社，2017.10
ISBN 978-7-121-32722-3

I. ①全… II. ①和… III. ①网页制作工具－程序设计 IV. ①TP393.092.2

中国版本图书馆 CIP 数据核字（2017）第 228608 号

责任编辑：田宏峰
印　　刷：三河市兴达印务有限公司
装　　订：三河市兴达印务有限公司
出版发行：电子工业出版社
　　北京市海淀区万寿路 173 信箱　　邮编：100036
开　　本：787×980　1/16　印张：16.75　字数：375 千字
版　　次：2017 年 10 月第 1 版
印　　次：2017 年 10 月第 1 次印刷
定　　价：68.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888，88258888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式：tianhf@phei.com.cn。

前　言

为何写一本全栈的书

为什么写一本以全栈为主题的书呢？这还得从我的工作经历说起。

在过去的十多年，我一直在从事与移动互联网相关的工作，从早期的手机软件开发到今天的移动应用，都离不开架构的支撑。在智能机出现之前，手机的软件架构群雄并起，各家手机厂商都在打造自己软件平台，直到 iOS、Android、Windows Phone 的出现，形成三足鼎立的时代。在经历了近五年的洗礼之后，进入移动互联网的巅峰时代。而今，iOS、Android 两大平台平分天下。

开发一款移动互联网产品，从表面上来说，似乎只需要做一个 APP，包括 iOS 和 Android APP；其实，如果想让上线的产品运营起来，就没这么简单了。通常，一个活跃度很高的产品，都是一款具有生态系统支撑的平台，它包括 iOS APP、Android APP、微信公众号、PC 网页、强大的后台管理，一个都不能少。如果采用传统的开发技术，打造这样的一款产品，需要组建一支十几人的开发团队，人员一多，沟通的成本可想而知。

移动互联网产品的一个最大特点是，一旦产品投放市场得到了用户的认可，其版本迭代更新非常之频繁。无形中，对团队的开发效率提出了更高的要求。

无论是 iOS 还是 Android，APP 原生开发模式的最大弊端是版本的迭代与升级的任务繁重。为了解决这个问题，才引入了 HTML5 的技术。从开发的技术工种来看，分为 APP（iOS、Android）工程师、前端工程师、后端工程师。这三个角色中，前端工程师直接面向终端用户，是产品的“门面”。如果后台选用 PHP、Java 之类的技术，前端工程师除了网页的制作之外，其他可做的非常有限，毕竟前端技术局限于 HTML、CSS 和 JavaScript。因为角色的分工比较发散，以致开发效率难以提升。为了解决开发效率和运维灵活性的问题，我们希望从前端寻求到一个突破口。

众所周知，前端工程师身怀三大法宝：HTML、CSS 和 JavaScript。这些前端开发语言既偏离 APP 的原生开发语言（Objective-C 或 Swift），又与后台的开发语言（常用的 Java）有着明显的差异。虽然 JavaScript 带有“Java”一词，但 JavaScript 与 Java 之间的关系如同雷锋与雷峰塔之间的关系，二者相去甚远。那么，有没有一种框架可以让前端开发人员“通吃”后台呢？

一个偶然机会，我接触到了全栈（Full Stack）的概念，并瞬间被它的理念所吸引。这里说的全栈，不是传统的 LAMP（Linux、Apache、MySQL、PHP），而是一种全新的以前端为主导的框架，所谓“大前端”、“全端”，就是指以前端为核心的框架。最终，我们把框架选型聚焦在 MEAN（MongoDB、Express、AngularJS、Node.js）上。MEAN 全栈技术框架所用到的每个组件（MongoDB、Express、AngularJS 和 Node.js），都是基于 JavaScript 开发语言的。原本 JavaScript 是为网页设计的语言，但自从有了 Node.js 之后，JavaScript 的春天来了，前端工程师也可以写后台了。Node.js 让前端开发像子弹一样飞！

开发一个产品之前，我们总会纠结应该选择怎样的技术框架。的确，框架的选型很重要，它直接决定了这个产品未来的走向，技术的选择需要考虑几个主要因素，其中包括自身所掌握的技能、软/硬件环境、生产环境的部署、产品上线后的运维等。

选用 MEAN 全栈技术，可以快速地实现敏捷开发，尤其是到了产品的运营阶段，其优势表现得非常明显。我们知道，今天的任何一款移动互联网产品，都离不开微信公众号的推广，大多出彩的产品，在它的微信公众号内，所展示的是一套完整的业务逻辑，而不是几个简单的页面。这就是说，一个运营成功的产品，对前端技术的依赖非常之高，更何况 APP 也可以采用混合开发模式（Native+HTML5）。

全栈工程师并不是“全能”工程师，它是通过一种全栈的框架，从繁重的技术中解脱了出来。正所谓：工欲善其事，必先利其器。这里的“器”，就是全栈框架，具体到这本书所推荐的，就是 MEAN 全栈框架。

践行全栈之路

用了 MEAN 全栈，它到底能带来什么好处呢？这里，以我们发布的一款产品——“点时”为例。“点时”APP 是一款轻量级的知识分享平台，以语音分享为主。这样的一款产品，从生态上讲，该平台包括：iOS APP、Android APP、微信、后台的课程发布与运维管理。传统的做法是项目开发组分为前端与后台两套人马，因为进度不一，要么前端等后端，要么后端等前端，而我们采用的是 MEAN 全栈架构，不再区分前端与后台，开发效率得到了明显提升。用了 MEAN 全栈框架，它带来的最大好处是减少了前、后端之间的依赖。

读者对象

这是一本讲述 MEAN 全栈入门的书，而不是一本从入门到精通的书。MEAN 全栈蕴涵的组件有多个，每一个组件都可以独立成书。书中的每一个知识点都是为后面章节中的实例铺垫的，泛泛的基础知识并不在本书讲解范围之内。

本书自始至终延续这样的一个主题：如何通过一种框架（MEAN 全栈），将前端和后台（端）贯穿起来，让前端工程师快速上手。

MEAN 全栈技术涉及的技术点很多，它是前端（Front-end）技术向后台（Back-end）的延伸。只有具备了前端的基础，才能更好地理解全栈架构的思想。如果尚未接触过 HTML、CSS、JavaScript，那么，有必要“恶补”一些前端的基础知识。

具体来说，这本书适宜的读者有：

想学前端技术的 APP（iOS、Android）开发工程师。随着 APP 多年的发展，APP 的优势和短板日益明显，原生技术无法解决的问题，需要前端技术（HTML5）来弥补，二者结合相得益彰，所以混合开发模式越来越受欢迎。如果一个 APP 开发工程师同时具备了原生与全端的技能，由“单翼”变成了“双翼”，其技术路线的前景将越来越广！

想学习后台技术的前端工程师。传统的互联网开发，分为前端和后台，在职场上也就出现了前端工程师和后端工程师。借助 Node.js 平台和 Express 后端框架，前端工程师可以无缝地延伸到后台技术。

如何阅读本书

既然是全栈技术，其蕴含的知识点无疑有多个方面。为此，本书分为入门篇、基础篇、实战篇。

入门篇：

这里没有讲述 HTML 的基础，也没有谈论 CSS 概念，而是直接切入 CSS 框架，一款主流的 CSS 框架——Bootstrap。在我所经历的互联网项目中，Bootstrap 是应用最为广泛的。这里还讲述了 JavaScript 特有的编程模式——函数表达式与函数式编程，在 Node.js 开发中，JavaScript 把这些特有的闪光点发挥得淋漓尽致。MEAN 全栈中所用到的数据交互格式和存储格式均为 JSON，学习全栈技术，必须掌握 JSON 的应用。

在入门篇中，没有讲述 jQuery 技术及其 AJAX，这是因为，在 MEAN 全栈中用到的 AngularJS 前端框架本身就兼具 jQuery 和 AJAX 的功能。

基础篇：

从这篇开始，我们将正式进入 Node.js 的世界。尽管 Node.js 功能很强大，但其生态系统的构建还要借助 Express、AngularJS、MongoDB 及模板引擎。

在市面上，我们会看到很多权威指南系列的图书，比如 Node.js 权威指南、AngularJS 权威指南、MongoDB 权威指南。这些书对每个专业技能都讲得很透彻，但很少谈及它们之间的关系。既然 Express 是基于 Node.js 之上的后端框架，对初学者来说，我们更希望在 Express 基础之上开发。

那么，为何要选用 AngularJS 呢？在吹响“全端”号角的今天，我们越来越强调前端框架的重要性。在前端的世界，AngularJS 可谓“玉树临风”。在 MEAN 全栈中，Node.js

和 Express 负责后端处理，而与网页交互的正是 AngularJS。因此，可以想象 AngularJS 在本书中所占比重之高。

关于 AngularJS，这里要特别说明一点：本书讲述的 AngularJS、示例中所引用的 AngularJS 均为 1.x 版本，具体来说是 1.4.6 版本。AngularJS 最新版本是 2.x。或许读者产生疑问，为何不用 AngularJS 最新的 2.x 版本呢？这是因为，它的 2.x 并不是在原有 1.x 上的升级，而是一个全新的版本，两者谈不上兼容之说。业内普遍认为，AngularJS 1.x 版本更成熟、应用更广泛、可参考的资料更多。在项目开发时，选择一个成熟的框架，十分重要！

把 MongoDB 数据库应用到 MEAN 全栈中，可谓相得益彰。通过 MongoDB，你会对全栈开发有一个完整的、全新的认知。

实战篇：

学习一门编程技术，最有效的途径还是实践。对于书中出现的每个知识点，都辅以相关的代码实例。每个篇章中的实例都不是独立的，而是沿用从易到难的线索。

实战篇演示了四个实例，每个实例并不是独立的，从知识衔接上看，是一环扣一环的。通常，一个完整的应用包括：数据与页面之间的绑定、网络请求、路由的分发、数据库的增删改查。我曾试着通过一个完整的应用讲述以上知识点，发现越到工程的后期越发臃肿，前后逻辑关系太复杂，以至于理解起来颇费周折。最终采取一个折中的方案：借用国外网站的经典 MEAN 全栈示例，在原示例的基础之上，对一些不易理解的地方，添加了补充的知识，正所谓“见招拆招”。

实战篇中示例，都是基于 MEAN 全栈的演练，只是侧重点有所不同，每个示例均附有完整的工程源码。

本书的源码

在学习本书示例代码时，可以按照书中讲解的步骤，一步一步地手工敲入所有代码，也可以下载本书的源码，本书所有的源代码都可以从 GitHub 下载。

源码下载地址：https://github.com/leopard168/MEAN-Full-Stack。

勘误和支持

我尽最大的努力确保正文和代码没有错误，但随着开发环境版本的变化，错误在所难免。如果读者发现书中的任何错误，如错别字或代码片段无法运行等，希望您能及时反馈给我。您提交的勘误不仅能帮助自己，还能让其他读者受益。

读者可以在下载源码的地方（GitHub）进行反馈，也可以通过后面的联系方式与笔者沟通。

致谢

参与本书编写的还有袁芳、和凌群、徐明志、胥方文、江美双、和凌云、马钧君、林志红、刘晓波。在本书成稿的过程中，我得到了很多人的指点和帮助，客套话不再讲太多。这里，特别感谢 MEAN 全栈的开源者，向开源精神致敬！每次赏析那些原创的示例，都能得到一次心灵的升华。

在本书出版之前，我曾以本书的书稿，给临沂大学本科生讲授全栈开发课程。令人欣慰的是，通过 MEAN 全栈框架，学生们很快完成了一个从前端到后端的项目。

感谢电子工业出版社，正是你们卓有成效的工作使我保持了敲击代码的激情。

作者交流方式：

作者的 QQ 及邮件：2385911707@qq.com。

作者的微信号：leopard2385911707。

作　者

2017 年 8 月

为什么选择 MEAN 全栈技术

概述

开发一个功能性的网站并不容易，它要借助很多种技术，需要一套组合拳，单纯的某一项技术是不够的。在描述网站构建时，常听到一个词语，这就是“技术栈”。比如，Linux、Apache、MySQL 和 PHP，把它们的首字母组合在一起，被称为 LAMP 栈。MongoDB 的工程师 Valeri Karpov 发明了一个缩略语 MEAN，指的是 MongoDB、Express、AngularJS 和 Node.js。的确，这是一个很不错的技术组合，而且读上去朗朗上口。MEAN 全栈（MEAN Full Stack）框架日益成熟，在网上可以找到大量的 MEAN 全栈示例。

如果想开发一个功能性网站，MEAN 全栈技术框架是一个不错的选择，但它不是唯一的选择。就拿数据库来说，即便基于 Node.js 开发，也不是非选 MongoDB 不可，用其他关系型数据库（如 MySQL）也是可以的；同样，作为前端框架的选择，也不见得必须用 AngularJS，用 Vue.js 也是可以的。这就是说，MEAN 全栈无法体现 Node.js 生态系统的多样性。MEAN 这个缩略词漏掉了一个重要的组件——模板引擎。模板引擎的类型有多种，我们完全可以通过手动方式来配置。

在 MEAN 这个缩略词中，毋庸置疑，其中无可替代的组件当然是 Node.js 了。作为运行 JavaScript 语言的服务端，Node.js 是其中的执牛耳者，尽管也有类似的服务端，但与 Node.js 比起来，难以望其项背。

起初，JavaScript 语言仅仅是为了编写网页，很难有其他用武之地。自从有了 Node.js，JavaScript 的春天来了。通过 JavaScript 这一项技术，把 MEAN 全栈技术贯穿在一起。

夸张一点说，学习 MEAN 全栈技术，只需要掌握一门 JavaScript 语言就够了。

什么时候用 Node.js

Node.js 是专门为 I/O 密集型操作和快速构建可扩展性的实时网络应用而设计的，比如说，一些网游、聊天系统等。通过 Node.js，你可以用最少的系统资源来服务大量的客户端，Node.js 就是为高扩展性而设计的。

对于搭建类似于 MongoDB 的文档数据库的 API 服务器，Node.js 也是一个不错的选择，

它可以将文档数据以 JSON 对象的格式存储在 MongoDB 中，然后通过 RESTful API 来操作它们。当从数据库读写数据时，不需要将 JSON 与其他类型的数据进行转换。

根据 Node.js 的结构特点，它不适用于 CPU 密集型的操作，Node.js 本身是一个运行 JavaScript 的服务器环境。

本书讲述的示例都是基于 Node.js 基础之上的应用，而不是 Node.js 架构。

MEAN 全栈简介

构建 Node 应用有很多选择，而 MEAN 全栈框架越来越成为一种趋势，MEAN 全栈主要由四项技术组成。

- MongoDB：用来存储数据的数据库。
- Express.js：服务器端用来构建 Web 应用的后端框架。
- AngularJS：用来构建 Web 应用的前端框架。
- Node.js：JavaScript 运行环境。

MongoDB 于 2007 年推向市场，由 MongoDB 公司运营。Express 最早由 T. J. Holowaychuk 于 2009 年发布，并已经发展成为 Node.js 之上的最主流的框架，它是一个开源的框架，社区活跃度很高。AngularJS 是一个开源的前端框架，它的背后支持者是 Google，到了 2010 年，AngularJS 已经被广泛应用，AngularJS 的发展势头强劲，从早期的 1.x 版本已经更新到今天的 2.x 版本。Node.js 是 2009 年发布的，Node.js 采用了 Google 的 V8 JavaScript 开源引擎。

通过 MEAN 全栈框架，可以将文档数据以 JSON 对象的格式存储在 MongoDB 中，然后通过基于 Node 和 Express 搭建的 RESTful API 来操作数据库，前端通过 AngularJS 构建的客户端来操作这些 API。AngularJS 通过 RESTful API 获取服务器数据后，再把数据交给前端模板引擎渲染，最终形成 HTML 页面展示给用户。要想完成这些操作，只需要使用一门统一的语言——JavaScript。这样一来，代码更加具有一致性和可维护性。另一个好处是，整个 MEAN 全栈技术所要处理的大多是 JSON 数据结构，而 MongoDB 中的文档对象也是 JSON 格式，通过 RESTful API 获取的后台数据也是 JSON 格式，正是这些一致的 JSON 格式，才省去了格式之间的转换，从而提高了开发的效率。

MEAN 全栈的四大组件关系

全栈开发包含了众多的知识点，可以说，每个知识点都可以独立编写成一本书。事实上，也确实如此。对于开发一个 MEAN 全栈应用来说，JavaScript 语言从前端贯穿到后台；数据以二进制 JSON（简称 BSON）格式存储在 MongoDB 中，基于 MongoDB 的 Mongoose 提供了类似 JSON 的接口，为操作数据库提供了极大的便利；源于 Node.js 的后端框架 Express 也是由 JavaScript 编写的；而前端框架 AngularJS 也是一个 JavaScript 库。MEAN 全栈的四大组件关系，如图 0.1 所示。

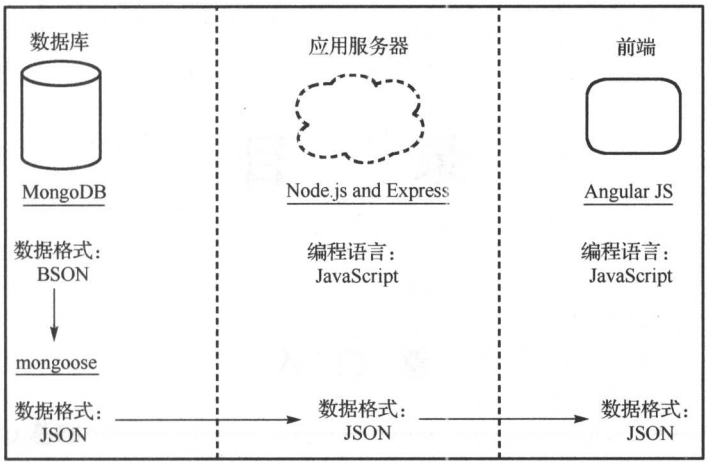

图 0.1　MEAN 全栈四大组件之间的关系

这里，再来回顾下 MEAN 全栈的技术组件：

- MEAN 全栈是由多种不同的技术组成的一个从前端到后台的框架；
- 在 MEAN 全栈中，选择了 MongoDB 作为数据库，从而凸显了 JavaScript 的优势；
- Node.js 与 Express 的“合体”，提供了一个完美的应用服务器框架；
- 作为前端框架，AngularJS 是那么地神奇，它把单页面应用和数据绑定发挥得淋漓尽致；
- MEAN 全栈技术为 JavaScript 提供了前所未有的平台，从而使得 JavaScript 成为了当今的一种主流开发语言。

目　录

入 门 篇

基 础 篇