Broadview®
www.broadview.com.cn

前端撷英馆

Meteor全栈开发

杜亦舒 著

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

前端撷英馆

Mo[ ]全栈开发

杜亦舒 著

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

内容简介

本书全面介绍了新一代全栈开发平台 Meteor。书中首先简要介绍了 Meteor 的概念和特性，然后通过各种示例讲解 Meteor 的用法，再用案例实践的方式综合运用所讲过的内容，加深对 Meteor 的理解，接着展示 Meteor 应用如何部署到生产环境中，最后探讨一些 Meteor 应用架构扩展的进阶话题。

本书面向对 JavaScript 全栈开发感兴趣的读者，可供希望快速进行产品开发和想尝试新技术的开发者参考。

图书在版编目（CIP）数据

Meteor全栈开发 / 杜亦舒著. —北京：电子工业出版社，2016.10
（前端撷英馆）
ISBN 978-7-121-29968-1

Ⅰ. ①M… Ⅱ. ①杜… Ⅲ. ①JAVA语言－程序设计 Ⅳ. ①TP312.8

中国版本图书馆CIP数据核字（2016）第229082号

策划编辑：张春雨
责任编辑：李云静
印　　刷：北京天宇星印刷厂
装　　订：北京天宇星印刷厂
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：17　　字数：304千字
版　　次：2016年10月第1版
印　　次：2016年10月第1次印刷
册　　数：3000册　　定价：75.00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：010-51260888-819　faq@phei.com.cn。

前言

这本书讲了什么

本书是一本 Meteor 的入门实践教程。Meteor 是新一代的 JavaScript（JS）全栈开发平台，基于 Node.js，但并不要求读者必须已经熟悉 Node.js。本书的目标是使读者阅读本书，能够理解 Meteor 不一样的技术思路，学会使用 Meteor 进行快速的 Web 开发，以及掌握对 Meteor 进行架构扩展的思路。

本书一共有 11 章，分别从入门介绍、功能讲解、项目实践、进阶拓展这 4 个方面对 Meteor 进行了阐述。

第 1 章和第 2 章为入门介绍，讲解了 Meteor 具体是什么，它的工作原理，以及 Meteor 的优势和不足。通过这两章的学习可使读者快速地建立起对 Meteor 的初步印象，然后详细讲解了 Meteor 的安装方法，读者从中可以体会到 Meteor 的快速与便捷。

通过前两章的入门介绍，读者已经大体认识了 Meteor，知道了它的特性，但头脑中还是会有很多问题，例如 Meteor 的开发方式有什么不同呢？ Meteor 的快速开发体现在哪些方面呢？…… 通过后面的深入讲解，这些问题就会逐渐被弄明白。第 3 章到第 7 章为功能讲解部分，将 Meteor 的知识结构拆分成几大块，逐一讲解模板的应用、MongoDB 数据库的操作方式、路由控制、用户系统的集成配置、与数据库沟通方式的优化和安全升级，并带有丰富的示例。通过这些功能的讲解与示例实践，读者已经达到可以开始实际应用 Meteor 进行开发的程度。

经过对 Meteor 功能模块的讲解与实践，下面便进入项目实践部分。第 8 章会以一个完整的项目为例，从头进行开发，综合运用前面讲解的各部分功能，从整体上

体会 Meteor 应用开发的全过程。

进阶拓展部分包括第 9 章到第 11 章，从功能开发阶段过渡到了产品上线阶段，分别讲解了 Meteor 应用中如何进行测试、如何把 Meteor 应用部署到线上产品环境，以及对 Meteor 应用在架构上进行扩展的方式，为应用的质量和应用的性能做好控制和准备。

如何阅读本书

Meteor 是 JavaScript 的全栈开发平台，所以阅读本书的基础要求是熟悉 HTML CSS JavaScript，但并不要求很深的熟悉程度。如果读者对这些基础知识不太熟悉，可以到 http://www.w3school.com.cn 网站上花费一点时间学习一下，只需要掌握基础知识即可开始学习 Meteor。Meteor 是基于 Node.js 的，不熟悉 Node.js 也完全没有影响；但如果了解 Node.js 的话，会有助于更好地理解 Meteor 的机制。

本书的风格偏于实践，从第 1 章就开始了动手实践，第 2 章介绍了环境搭建和项目创建的方式，并推荐了 Meteor 开发所需要的工具和资料，后面的章节中都包含了大量的示例代码。所以，强烈建议跟随书中的实践步骤和代码进行亲自操作。因为实践是学习新技术的最好方式，实践可以让我们快速掌握对新技术的应用，也可以加深对技术特性和理念的理解。在实践过程中会遇到各种问题，对问题的思考和解决过程就是非常好的学习过程。

本书的优势

- 轻松入门。本书以 Meteor 的发展历史、核心优势为切入点，详细讲解了 Meteor 的优势与不足、工作原理、功能开发、进阶技术等，内容由浅入深，便于快速入门。
- 上手容易。本书的各个章节都集合了丰富的实例，尽可能地结合实际开发中常用的场景，让读者快速上手。在讲解完 Meteor 的各个局部知识后，特意安排了一个实践项目，综合运用了各部分知识，便于读者巩固前面所学到的内容。
- 架构扩展。本书的最后一章单独讨论了 Meteor 应用的架构扩展，结合 Meteor 应用的特性，给出相应的架构扩展建议，为实际 Meteor 项目的壮大做好准备。

目录

第1章 Meteor简介

Meteor 是一个新兴项目，热度极高，广受开发人员的喜爱，非常值得学习。本章我们一起走进 Meteor 的世界，看看 Meteor 具体是什么，它有什么特点，它为什么会如此流行，以及它的优势和弱势，并实际上手体验，对 Meteor 做一个大概了解。

1.1 Meteor是什么

Meteor 是一个开源的全栈 JavaScript 开发平台，构建在 Node.js 和 MongoDB 之上。全栈开发平台已经有不少了，Meteor 有什么特色呢？

“meteor”这个单词的意思是“流星”，流星的特点是快，一闪而过；同样地，Meteor 的特点就是快，目标是为开发者提供一个快速开发的平台。

虽然 Meteor 是一个很年轻的项目，但因其开发速度快而闻名，受到大量开发者的喜爱，GitHub 上的 star 数量已达惊人的 33000+，与 Linux 之父 Torvalds 创建的 Linux Kernel 项目相当。

Meteor 这个项目的来源非常有趣。Meteor 的几个创始人本来是要做一个在线旅游点评网站，并且已经进入了著名孵化器 YC，准备开干了。

但在筹备过程中，和孵化器的其他伙伴聊天时，发现大家都有一个共同的问题，

就是开发效率不高，常常需要做很多重复性的工作。

所以他们改变了创业想法，决定做一个开源的开发平台，提供一套完善的基础功能，减少重复劳动，提高开发速度，并希望有桌面应用一样的顺滑体验。

说干就干，他们在 2011 年 10 月 1 日推出了 Meteor 预览版，仅仅在 8 个月之后，Meteor 就得到了 IT 大佬们的投资。

1.0 版本发布之后，在 GitHub 上就进入了 top 20，成为当时第 11 位的流行项目。

Meteor 现在已经发展成了一个生态。因为基于 Node.js，所以其本身就可以受益于 Node.js 的庞大资源，而且 Meteor 自身也是社区模式，扩展包数量不断增长，内容已经极其丰富，功能覆盖面非常广。Meteor 生态在健康、快速地成长。

1.2 Meteor快速起步

1.2.1 创建新应用

从安装 Meteor 环境开始，一直到把测试项目运行起来，一共需要几个步骤、多长时间呢？我们一起来实验一下。

Meteor 支持 OS X、Windows、Linux。以 OS X 和 Linux 为例，在命令行执行以下命令来安装 Meteor：

```
curl https://install.meteor.com/ | sh
```

运行完成后，创建一个新项目：

```
meteor create myapp
```

运行项目：

```
cd myapp
meteor
```

访问项目，打开浏览器输入 URL：http://localhost:3000/，页面效果如图 1.1 所示。

这时，项目已经运行成功，并有默认的示例功能，单击“Click Me”按钮后，提示信息中的数值自增。

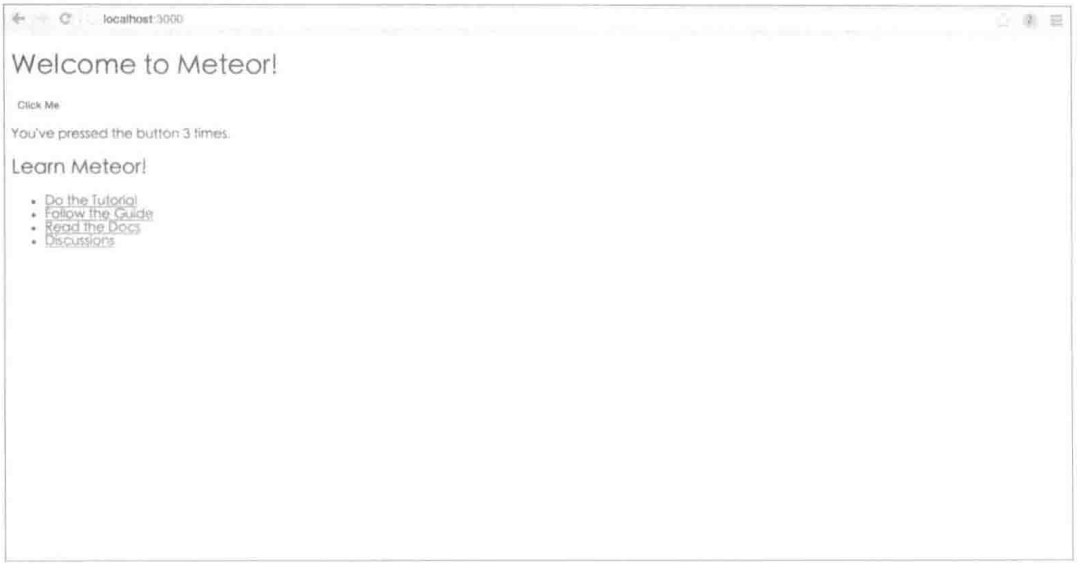

图 1.1 新建项目的访问效果

1.2.2 与 LAMP 对比开发过程

经过对 Meteor 的初步体验，是不是感觉很快？下面我们回顾一下常规开发平台的起步过程，以 PHP 为例。

（1）环境的安装

下载并安装 Apache/Nginx、PHP、MySQL。

（2）安装开发框架

直接用 PHP 写页面、操作数据库、控制路由等不现实。选择一个功能全面的框架是必需的，选择后下载并安装，再配置数据库的连接信息。

（3）引用 JS 库

一个好用的 JS 框架或者库是 Web 开发中必需的，例如 JQuery，要将其下载到项目目录中。

（4）代码开发

假设要实现上面 Meteor 项目中的默认功能：单击按钮后，页面中的值加 1 后自动更新。需要在页面中引用 JQuery，监听按钮的单击事件，处理函数中获取页面中现在的值，执行加 1 计算，然后更新 DOM 内容。

可以感受到这个过程并不简单。第 1 步是比较复杂的，即使现在有成熟的一键安装工具，但耗时也较长，而且在真正的线上运营环境中使用的话还需慎重。在第 4 步的代码开发中，页面内容的更新都需手动处理，如果涉及数据库操作则会更复杂一些——需要定义表结构、开发数据库的模型代码、把数据库操作结果传给前端展示（可能还需要使用 JS 来处理数据）。

我们再看一下 Meteor 的开发过程。

（1）安装环境

一键安装，无须任何其他配置。

（2）创建项目

通过 meteor create 命令创建出一个新的项目，无须安装其他框架和 JS 库，也不用配置数据库。

（3）代码开发

默认代码中的思路是：页面中引用动态值，JS 监听按钮的单击事件，事件处理函数中对变量值进行加 1 计算，无须操作 DOM，页面自动更新。

代码很简单，看一下 client/main.html 和 client/main.js 就可以大致明白其思路和做法了。

虽然在默认代码中没有涉及数据库，但数据库操作也很简单，直接用 JS 处理 JSON 结构的文档对象，插入的数据和查询结果都是 JSON 结构。

通过简单比较，Meteor 的开发过程确实非常便捷，因为 Meteor 帮我们做了很多幕后工作。而且通过后续的介绍和实践你还会发现 Meteor 更多强大之处。

1.3 Meteor 的工作原理

1.3.1 工作流程

Meteor 在工作方式上进行了较大创新，和传统 Web 应用区别较大。下面先回顾一下传统应用的工作流程，如图 1.2 所示。

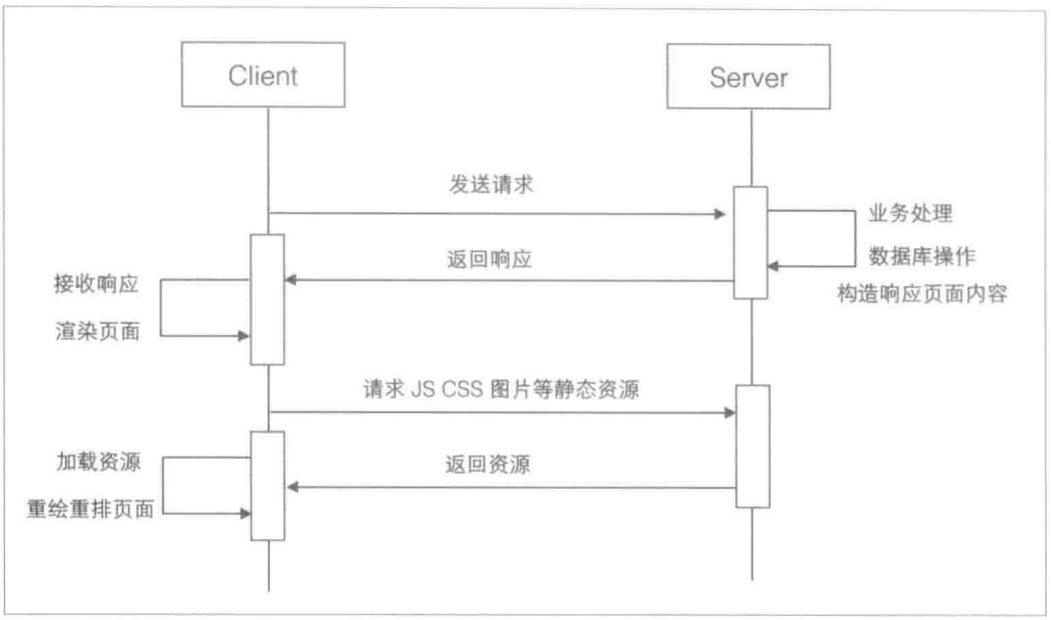

图 1.2　传统应用工作流程

客户端（Client）负责向服务器请求所需的数据、资源，然后渲染显示；服务器端（Server）负责业务处理、数据库操作、构造响应内容、资源管理，服务器端的责任大、任务重。其各自职责关系如图 1.3 所示。

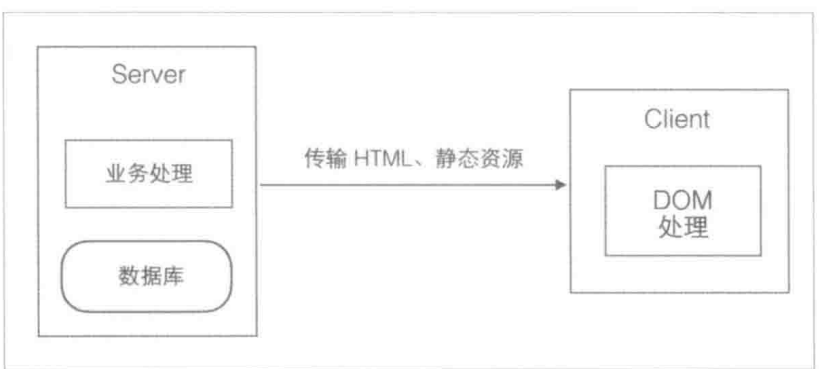

图 1.3　传统应用服务器端与客户端的主要职责

Meteor 的工作方式更像是手机 APP。客户端首次访问 Meteor 应用时，会从服务器把需要用到的资源都加载到客户端，如 JS、CSS、字体、图片，并创建一个 mini 数据库。然后和服务器端建立好数据通信的通道。之后，用户操作应用过程中涉及业务操作时，也是在客户端进行处理；进行数据库操作时，也是操作客户端的

mini 数据库。服务器端只负责向客户端传输数据、数据的安全写入，以及执行一些只能在服务器端进行的操作，例如发送 email，如图 1.4 所示。

图 1.4 Meteor 中客户端和服务器端的工作流程

Meteor 应用的客户端包含了应用所需的静态资源、业务处理代码、一个简化的数据库。如手机 APP 一样，很多操作直接在本地完成，需要执行特定动作和需要数据时才请求服务器端。

所以相比较于传统 Web 应用，Meteor 选择了重客户端、轻服务器端的模式，充分利用现代客户端强大的运算能力，减轻服务器端的压力。

1.3.2 核心技术

Meteor 的工作方式必然需要一些特定的技术来支持，让我们来了解一下 Meteor 的几个核心技术。

1. mini 数据库（mini-database）

Meteor 的底层技术中首先吸引我的就是客户端的 mini 数据库。Meteor 目前支持的数据库是 MongoDB，所以客户端的 mini 数据库就是 miniMongo。

对于开发人员来讲，miniMongo 就像是一个真实 MongoDB 数据库，可以进行各种增删改查的操作，和 MongoDB 的 API 完全一致。

miniMongo 的主要作用是缓存数据，相当于服务器端数据库的局部镜像，它不会缓存全部数据，只是缓存当前客户端用到的数据。

使用 miniMongo 的效果就是应用运行非常快，而且提供了更好的用户体验。例如用户保存了一条数据，Meteor 会先保存到 miniMongo，保存成功后立即反馈给用户，

体验极其顺畅；同时 Meteor 会把数据同步到服务器端的真实数据库中，这个过程对于用户和开发者都是透明的。

那么如果网络出现问题，或者后台数据库操作时出现问题时，数据没有同步成功怎么办？

当客户端发现没有同步成功后，会通知用户出现了问题，页面执行相应的错误处理逻辑。例如用户保存了一条数据，数据先被写入 miniMongo，然后反馈用户操作成功，同时后台进行数据库同步。万一服务器端操作失败，会通知客户端，客户端会告知用户之前的操作有问题，并执行相应的错误处理流程。

2. Tracker

Tracker 提供了响应式应用的基础功能。下面先简单了解一下什么是响应式。以之前创建的项目为例，页面中有一个按钮，单击按钮后，页面中显示的那一个数字自动加 1。通过查看代码，代码的逻辑如图 1.5 所示。

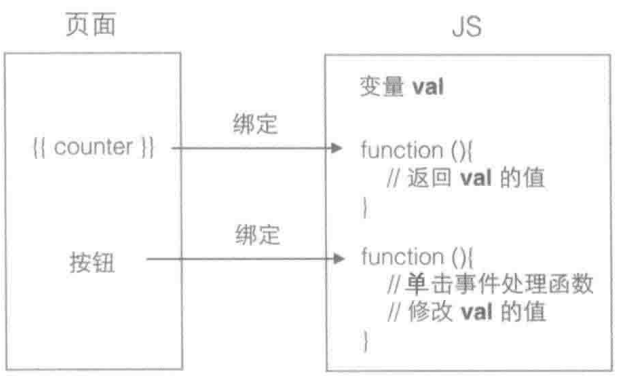

图 1.5　示例代码逻辑

{{ counter }} 通过函数关联了 val 变量，按钮单击事件的处理函数中修改了变量 val 的值，并没有更新页面中的内容，但 {{ counter }} 自动更新了，这就是响应式。

响应式的背后技术基础就是 Tracker。Tracker 会跟踪目标数据，当其有任何变化后，都会重新计算使用到目标数据的地方。

在上面的示例中，变量 val 是一个响应式变量，会被 Tracker 跟踪，{{ counter }} 是变量 val 的消费者，当 val 被修改后，Tracker 便通知它的消费者进行更新。

3. DDP

DDP 是一个数据传输协议。Web 应用通常会使用 HTTP，为什么还要使用 DDP