# 序

若无辞令，何以言之。

超媒体（hypermedia），可表现为其元素描述自身形态以及与其他元素之间联系的一组能力。万维网（World Wide Web）的发展正是由超媒体所驱动的。超媒体技术不仅仅可以用于建设易于使用的web站点；它更是一种新的分布式计算形式，强大而又灵活。

Web技术和超媒体技术并不是什么新鲜事。甚至早在20世纪90年代中期，我们便可以把超媒体技术应用在分布式计算中了。然而，在很长一段时间里，我们仅仅是作为“消费者”来使用超媒体的。超媒体是一个非常易于理解的直观概念——我们都使用Web，而要从发展的角度来理解它，就不是那么容易了。

我们的困难源于概念障碍。早在我们形式化描述Web架构之前，Web却早已深入到我们的日常生活中了。我们花费了二十一世纪循序渐进地帮助开发者对Web的力量达成共识——Web那一直存在着的力量。

在我看来，本书中所描述的超媒体将是自Roy Fielding首次定义REST架构以来的一次概念上的革新。Mike Amundsen使用了总称“超媒体”，并分别研究其意义（能为我们带来什么）以及其工作原理。

是什么导致了某些数据格式只对特定的应用有意义？为何HTML是如此的通用，甚至对于非消费型应用也是如此？它的缺陷又在于何处？在Mike关于超媒体的见解中，这些问题都有着确切的解答。而这些解答（超媒体）正是我所期望的即将成为下一代Web服务和Web技术的基础。

Mike不仅仅描述了超媒体是什么，更直观地向我们展示了超媒体是如何工作的。

——Leonard Richardson

2011年11月

# 前言

*When you set out on your journey to Ithaca,*

*pray that the road is long, full of adventure,*

*full of knowledge.*

*- Constantine P. Cavafy*

当你开始前往Ithaca的旅程

但愿你的旅途漫长

充满奇遇

充满知识

——Constantine P. Cavafy

## 超媒体API设计

本书的主旨在于设计超媒体API。对于一些读者来说，这可能有一些陌生。当今已经有非常多的书籍讲述编程语言、数据存储系统、Web框架等等。而这本书并不是这样的书。相对地，这本书覆盖客户端与服务器间传递的消息的本质，并探究如何改善并提高这些消息的内容和价值。我个人觉得，这是一个多么刺激而又迷人的领域！

正如此类作品的发布趋势所表现的那样，当今Web开发中对API的依赖似乎正日益增多。总的来说，这是件好事情。这正意味着在越来越多的开发者中，这样的观念正日渐深入人心：万维网不仅仅是一个分享数据的好地方，更是一个分享服务的好场所；而这正是Web在倡导之初的根本目标。

尽管如此，我相信Web API的爆发可能会将我们带入一个及其艰难的境地。在过去的一些年里，我见过太多的案例使用源于桌面应用或者是本地网络应用的概念而设计的API进行实现，当部署到万维网层级上，这些应用表现得很不理想。随着时间的推移，这些解决方案越发表现出其脆弱性，使得开发者在逐渐复杂和昂贵的维护面前显得心力交瘁。简而言之当今Web中符合超媒体概念的API并不多见。

本书旨在帮助新设计并上线到万维网上的API更加易用，并且更能经得起时间的考验。帮助这些API更能担起挖掘超媒体链接价值的责任。当下，HTTP协议族的编纂、HTML的实现、Atom/AtomPub以及其他本地超媒体格式仍然是驱动web发展的主要动力。

## 本书面向的读者

本书的主要目标是增加超媒体内容在网站上的使用数量，更在于提高其质量。因此，本书面向多种不同的读者。

首先，本书为系统架构师提供了指导。希望本书能够有效指引那些基于超媒体的应用，使它们更具稳定性和可优化性。当超媒体成为系统架构的一个不可或缺的部分时，他就为系统架构师提供了很多种可能。本书将阐述架构师通过把超媒体数据视为关键架构构件（而非仅仅是一个由客户端和服务器导入的有效负载）来提高系统的稳定性和灵活性。

其次，本书为那些负责实施客户端和服务器任务的读者提供了很多宝贵意见，并以实例说明如何处理超媒体信息本身。到目前为止，关于网站实施的书籍大都关注于服务器在处理超媒体方面所起的作用。作者认为这种想法往往导致客户端实现不当，常常忽视甚至否定超媒体信息在网络中的价值。超媒体作为架构支柱的主要优势就在于它鼓励客户端“以媒体形式编写代码”，而不是编写一些将消息视为简单数据的应用程序。超媒体客户端编程是一种技巧，需要我们花时间去掌握它。尽管本书并不是专门关注于超媒体客户端编码，作者希望本书能向读者展示的例子和优势能够推动更多有能力的人加以实践和利用以实现更多机遇超媒体优势的技术。

## 本书不会详述的概念

尽管本书内的实例使用HTML5、Node.js、CouchDB，但本书并不是这些技术的入门书籍。机敏的读者还可能发现本书作者在使用这些技术时并不专业。因为作者并不想从专家级的角度来使用这些技术。相对地，在本书中这些技术仅仅作为工具用于阐述设计和实现超媒体的关键点。本书附录列出了在这些技术领域中一些非常优秀的参考书，想深入了解这些技术读者可以去阅读这些更为专业的书籍。

这本书并不涉及HTTP协议的标准以及相关的web标准。本书附录中列举了本书成文所参考的一系列非常有价值的RFC文档和其他的标准文档。另外读者还可以在附录的推荐书籍中找到很多非常值得一读的好书。

最后，在REST（Representation State Transfer表述性状态转移）崛起的当前背景下，本书并不准备深入探究这个话题。确切地说，超媒体是REST非常重要的一个方面。关注超媒体本身的设计能够很大程度上提高包括REST在内的很多分布式网络架构的质量和功能性。想更深入地了解Fielding提出的REST的读者可以参看本书附录中给出的推荐书目。

## 本书的主要内容

本书的书写结构允许读者从任何一个感兴趣的章节开始阅读，你并不需要一章一章按次序地阅读本书。本书中有很多相关章节指引链接引导读者找到他们在阅读过程中可能被跳过的内容。希望这样的设计能够帮助读者日后更方便地参考本书。

本书的内容安排如下：

***第一章：理解超媒体***

这是本书的一个概念性的章节。首先给出一系列历史上已经存在的超媒体的示例，如HTTP及HTML。然后在更多的超媒体的情形下，进一步分析当前超媒体的设计，给出一系列设计新的超媒体载体的方法。

***第二、三、四章：实战超媒体***

中间的这些章节以实例展示了实现一个具有较完整功能的超媒体的过程。这些章节指引读者体验超媒体实现的全过程。从应用场景的考量，到设计元素的选取，再到样例数据的创建，最后到实现一个完整的符合用例需求的解决方案（包括服务端和客户端）。尽管示例看起来相对简单，但这囊括了绝大部分在真实的产品级的解决方案设计中可能会遇到的情境细节。

***第五章：标准化超媒体***

这是本书中的一个总结性章节。本章给出了将超媒体标准化的建议，并建议如何将这些标准新增注册到现有的标准中，例如IANA、IEFT和WC3。本章还涉及超媒体概念标准的版本管理和超媒体类型的扩展管理， 这与一些优秀的API和超媒体设计中的版本管理和扩展管理是类似的。

***附录***

本书还有了一个很大的附录。附录中包括本书所引用到的参考资料，还包括本书中没有提到的一些相关的扩展内容。在读者读完此书后或者在将来的实际应用中，这些资料可能会具有更有意义的参考价值。