

现代API

通往架构师之门

李 泉 著



清华大学出版社

现代API

通往架构师之门

李泉 著



清华大学出版社
北京

内 容 简 介

本书首先回顾系统集成及服务的历史,对其核心概念和核心思想进行重新阐述;然后从基本概念、REST 架构、生命周期、具体实施、最佳实践、业务影响和技术前瞻等方面对 API 进行全方位的介绍;最后是作者对如何做一个好的架构师的感悟与建议。贯穿全书的是作者在近 20 年里,为北美 18 个行业里的 50 多家大型公司进行系统集成及 API 项目设计和实施积累下来的实战案例。

本书为有志于成为系统集成和 API 架构师的程序员提供了一条学习和提高的路线图,适合程序开发人员及管理人员阅读和参考。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

现代 API:通往架构师之门/李泉著. —北京:清华大学出版社,2018
ISBN 978-7-302-50569-3

I. ①现… II. ①李… III. ①程序设计 IV. ①TP311.1

中国版本图书馆 CIP 数据核字(2018)第 141285 号

责任编辑:薛 慧

封面设计:何凤霞

责任校对:赵丽敏

责任印制:杨 艳

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印装者:三河市国英印务有限公司

经 销:全国新华书店

开 本:170mm×240mm 印 张:20.75 字 数:362 千字

版 次:2018 年 8 月第 1 版 印 次:2018 年 8 月第 1 次印刷

定 价:68.00 元

产品编号:079480-01

1990年，多数企业里的信息系统和应用都是各自为政、互不连接的。大家开始意识到“信息孤岛”对业务发展的制约，并开始致力于不同系统之间的信息共享。

2000年，有系统、成体系的企业应用集成（EAI）开始出现和完善。这一阶段关注的重点是如何在通信机制和通信数据两个层面上利用企业服务总线（ESB）将需要进行集成的系统和应用之间的关联性进行解耦的问题。事件（Event）和公共数据模型（CDM）便是这个阶段相应引入的概念。然而，对涉及系统集成应用的开发者来说，解决每一个独特的系统和应用连接的复杂性问题依然是最大的挑战之一。

2005年，服务的概念开始被广泛采用。一统天下的SOAP服务逐渐成为每一个开发员必备的技能。服务主推的是系统连接和系统所包含的业务流程的标准化，将系统连接的复杂性（对系统集成设计和实施）掩盖起来，强调的是业务逻辑的重用性。

2015年，早在2000年就已提出的REST的架构风格被越来越多地认可，REST API大行其道。API并非要取代系统集成和服务；而是要在更高的抽象层次上对业务进行重新思考，力图将企业的资源放到业务应用开发团队的手中，让他们能够自主、自助地完成自己的业务项目，并最终实现应用互联网和成规模的API经济。

本书首先回顾系统集成及服务的历史，对其核心概念和核心思想进行重新阐述；然后从基本概念、REST架构、生命周期、具体实施、最佳实践、业务影响和技术前瞻等方面对API进行全方位的介绍；最后是作者对如何成为一个好的架构师的感悟与建议。贯穿全书的是作者在过去近20年里，为北美18个行业里的50多家大型公司进行系统集成及API项目设计和实施积累下来的实战案例。

期待本书能够为有志于成为优秀的系统集成和API架构师的开发员们提供一条学习和提高的借鉴之路。同时，也希望本书（尤其是第14和15章）可以为正在寻找企业创新和数字化转型方面商业案例的MBA教员和学员们提供一些思路和素材。



李 泉

资深系统解决方案架构师，目前任职于美国MuleSoft软件公司（已被美国Salesforce公司收购）咨询服务部。曾任职于美国TIBCO软件公司咨询服务部。在过去18年里曾领导和参与了多个世界上很有影响的大型企业应用软件系统的设计和开发项目，用户包括北美18个行业里的50多家大型公司，主要针对最新的大型企业软件系统、API及云平台应用的设计和技术实施。

在此之前，曾在美林证券公司纽约总部任副总裁，负责面向服务架构的应用软件项目的标准化工作，并监督辅助项目的标准化执行（SOA CoE）。

长期以独立IT咨询顾问的身份从事大型企业应用软件系统的服务工作，主要侧重于企业应用整合（EAI）、面向服务架构（SOA）的软件开发、现代API应用、实时复杂事件处理（CEP）以及大数据分析等。曾与美国电力科学院及斯坦福大学联合进行大型电网模型处理和系统整合的研究工作，并在IEEE发表论文。

拥有美国休斯敦大学电子与计算机工程硕士学位和工商管理硕士学位，清华大学化学工程学士学位。并拥有Enterprise Java Architect和Java Programmer证书，MuleSoft Certified Developer证书，以及TIBCO Certified Professional证书。

2017年在清华大学出版社出版译著《重塑IT：应用互联网如何改变CIO的角色》。

1991

For Robin, Lindsey & Laura

序言

从 2018 年开始,绝大多数的中国企业都会把数字化转型作为企业核心战略;2020 年,中国 GDP 的 20%将来自数字化转型的增加值。在新的时代,企业应对数字化转型的速度,以及创造数字化产品、服务和体验的能力,不仅将决定其未来的发展,还将决定其未来的生存。这不是一件企业想不想做的事情,而是不得不做的事情。即便你已经是行业领军者,依然有可能被来自行业外部的颠覆者改变游戏规则而走向衰败。这样的例子比比皆是。

企业数字化转型成功的关键不仅仅在于数据有多少,更重要的是应用数据和信息的能力。而能够正确应用数据和信息的前提,是能够将原始数据以合适的形式和语境,在合适的时间呈现给合适的应用。这就是系统集成需求的根本由来。

为了应对企业数字化转型过程中遇到的数据格式不统一、内容不一致,调用方式五花八门,系统竖井林立、孤岛丛生,以及内外部协作难、创新慢、流程长等一系列的挑战,势必要有一种新型的技术架构来满足企业业务越来越快、越来越敏捷的要求。

REST API 并不是一项新的 IT 技术,但本书赋予了基于 API 开发、集成的新的应用场景。其中对于 API 的三层架构的阐述尤为独到:以系统资源 API打通系统孤岛,发现并呈现统一格式的数据资产;以协同流程 API根据业务逻辑对数据资源进行组合、提炼,实现功能协作和流程重组;最终以用户体验 API用于业务应用的快速、自主自助式的开发,推动能力输出和业务创新。

基于现代 API 的架构思路,企业可以更好、更快地将内部资产理清和重组,进而对业务流程、现有资源深挖和商业模式重新进行思考和创新。作者在这方面列举了大量亲身经历的案例,并对其中深层的架构思想和实施步骤细节、最佳实践、相应的组织结构安排等都进行了较为系统的论述,既有理论性,又有可操作性。白山云科技公司在业务工作中同样遇到过大量类似的客户案例。仅举一例:为建设统一的船舶产业链云平台,中国某大型船舶物资供应集团公司的 10 多个业务厂商进行各自的子系统建设。在每一个厂商依据自身的标准进行了系

统建设之后,发现各子系统之间的通信较为困难。如果以 ESB 来实现各业务系统交互,实施过程中各开发厂商就需要大幅改造自身系统匹配 ESB 的 Web Service 服务,这就会存在影响业务系统稳定运行的风险。另外,各业务系统通过 ESB 进行交互,相互之间信息传输无法直观展现和实时掌控,出现问题时排查比较困难;也无法将平台中的业务内容以标准化的方式对外界开放以及进一步建设完善的业务资源生态系统。

为保障快速、稳定地实现信息交换和资源共享,白山云科技公司参与主持的这个项目中的各业务系统将已开发的各种格式的接口统一转换成 REST API,并以 REST API 的方式实现系统敏捷交互,快速响应业务要求的变化;真正在 API 上线、安全、监控、分析、编辑、运行、权限控制、流量控制、告警、下线等流程上实现了 API 的完整的生命周期管理。基于这样的敏捷集成的云平台,今后可以将各种新业务和应用进行快速编排、测试并上线;还可以将所有封装好的系统 API 和流程 API 呈现在 API 市场中,对企业内外进行能力开放,供上下游合作伙伴调取、计费,实现数字化推进,形成全新的船舶生态和 API 经济体。

各方面的数据显示,北美的企业在系统集成、资源共享,进而持续创新和数字化转型这一块的需求价值在千亿美元的量级。国内企业相同的需求也是巨大的,随着企业持续创新和数字化转型的扩大和深入,这种需求只会加速增长。然而,尽管需求和市场巨大,白山云科技公司在开发和销售自己特有的系统集成和 API 平台的过程中深刻地体会到,满足这方面要求的人才尤其是领军人物极其缺少;相关的技术资料、信息、案例也十分匮乏。作者的这本书理论和实践并重,恰好填补了国内 IT 图书市场在这方面的空白。

作者真诚地向有志成为企业级 IT 解决方案架构师并推动企业持续创新和数字化转型的综合型 IT 人才推荐这本书。

赵鹏(Eddie Zhao)

白山云科技有限公司 产品副总裁

2018年4月26日于北京

前言

2018年5月2日,在世界范围内(包括在中国)享有盛誉的客户关系管理(CRM)软件供应商 Salesforce 以 65 亿美元的价格完成了对系统集成和 API 管理平台供应商 MuleSoft 的收购,其收购价格达到了 MuleSoft 股价溢价 36% 以上,为 2000 年 .COM 泡沫以来收购溢价的最高。关于 Salesforce 收购 MuleSoft 的具体原因,在网上可以看到各种各样的分析。有一点是确定的:资本市场愿意为能够有效解决系统和数据整合问题的方案、平台和技术付出高昂的代价,并期待丰厚的回报。

尽管很多人都认为系统集成是一个老话题,REST API 也是耳熟能详的技术,没什么新东西好谈了,然而,很多财富 500 强公司(甚至 50 强公司)在一二十年后依然在寻找好的系统集成平台和集成框架;大多数使用 REST API 的技术人员却对 REST 的架构风格以及围绕 REST API 的愿景知之甚少。这些都从一个侧面反映出,我们对围绕系统集成和 API 的基本概念和基本原则并不十分清楚,而且常常不能够准确清晰地进行表述。

写作本书的想法至少有两三年的时间了。但在很长的一段时间里,作者自觉很难做到几句话就清晰地回答书稿出来之后必然要面对的一个问题:“你到底想说什么?”是的,一本书没有明确的重点就很难具备贯穿全书的一致性,不仅内容会零散,也没有办法吸引来所希望针对的读者群。

在细细地梳理各种思绪之后,作者发现了两条主线:一条是关于“事”的,另一条是关于“人”的。

在“事”这条主线上,作者在过去近 20 年里,作为 MuleSoft 和 TIBCO 两家软件公司的资深服务解决方案架构师主持和参与了北美 18 个行业、50 多个客户的大型系统集成、面向服务架构和现代 API 项目。客户中的绝大多数是财富 500 强,甚至是 50 强的企业。中间经历了相关的 IT 技术翻天覆地的演变和发展,积累了很多的体会和感悟。

在“人”这条主线上,在近 20 年的时间里,作者不仅在项目上经常地与客户及合作伙伴进行技术方面的讨论,还为公司进行了 200 多次的解决方案架构师

人选的技术面试,并在公司里“带徒弟”,为年轻的架构师们作指导(Mentor)。这样的经历让作者有机会对架构师的学习和提高过程有了一个同时具备空间维度(不同的人)和时间维度(同一个人的成长经历)的观察和思考,希望从中能够总结出一份架构师的学习路线图。

在技术的主题中,本书的重点是现代 API。从来没有接触过现代 API 的读者可以用任何浏览器去访问 <https://www.pokeapi.co/>,应该看到类似下面的口袋妖怪(Pokemon)信息内容:

```
{
  "id": 1,
  "name": "bulbasaur",
  "base_experience": 64,
  "height": 7,
  "is_default": true,
  "order": 1,
  "weight": 69,
  "abilities": [
    {
      "is_hidden": true,
      "slot": 3,
      "ability": {
        "name": "chlorophyll",
        "url": "http://pokeapi.co/api/v2/ability/34/"
      }
    }
  ],
  .....
}
```

简单地讲,现代 API 就是用最普遍的 HTTP(s)的通信方式对世界上的任何(信息)资源进行调用。如果你对面向服务架构(SOA)和 SOAP Webservices 十分熟悉,而对为什么会出现 REST API 心存疑惑;或者你已经在使用 REST API,却无法用不超过三句话来说明服务与 API 之间在架构思想上的区别,那么这本书就是为你准备的。从大家都熟悉的 SOAP Webservices 到 REST API,这背后的架构理念、项目实施方式和相应的企业组织结构以及 API 对企业业务深远的影响,还有一个开发员如何在这个学习和实践的过程中成长为一名合格的架构师,这些就是本书想要说清楚的话题。

如果在这两条主线中再找出一条共同的主线,那应该就是“分享”。关于“架构”和“服务”的话题在市面上已经有太多的书了,而关于现代 API 的书在北美图书市场上也已开始出现。但是现实往往是很残酷的:既然人人都知道点对点集成做法的害处,那为什么来自世界著名的软件公司和著名的 IT 咨询公司的 API 架构师们在一个客户项目上花掉几千万美元服务经费(不包括软件授权的费用)后,最终的结果还是点对点集成的失败呢?为什么在项目设计的过程中说服别人那么难,而项目结束时能够承认错误、总结经验教训也那么难呢?任何事情都是这样,说到是一回事,能做到则是另一回事。理论要经过实践的摸索,包括经历失败,才能转化为成功的结果和经验。

现代 API 是当前很多热门技术和企业战略的使能者。云计算、数据共享、应用网络、企业数字转型与创新、智能城市、物联网,等等,在这些热门话题中都能找到 API 的身影。一个现代 API 架构师的最高境界就是在深入理解 API 设计和开发工作的技术细节之上,对 API 战略在企业转型和创新中的作用具有深刻的洞察力,并对企业业务的提升产生真正的影响。

对于希望自己开发出自用的或是商用的各种 API 工具的公司和架构师来说,本书,尤其是第 9 章和第 10 章的内容,也具有重要的借鉴意义。

作者期待与读者一起探索,共同提高! 欢迎读者与作者进行技术及各方面的交流并提供意见和反馈。请发邮件至 charlesquanli@gmail.com。在这个学习过程中的乐趣和成就感才真正是妙不可言!

李 泉

2018 年 5 月 18 日于美国休斯敦

目录

第1章 概述	1
1.1 什么是架构和架构师	1
1.2 这本书是为谁写的	2
1.3 为什么写作此书	3
1.4 通往架构师之路的路线图	5
1.5 架构师应该具备的素质	7
1.6 对架构师的学习和培养过程的几点建议	7
1.7 本书的主要内容	8
1.8 总结	10

第1部分 基础篇

第2章 重新看待系统集成	13
2.1 系统集成历史的快速回放	13
2.2 到底什么是系统集成	15
2.2.1 系统集成之信息更新	16
2.2.2 系统集成之信息组合	18
2.2.3 系统集成之连锁行动	19
2.3 系统集成的技术组成部分	24
2.3.1 BUS——高速公路	24
2.3.2 连接器——高速公路的进出口	25
2.3.3 CDM——高速公路运输的集装箱	26
2.3.4 数据转换——运输过程中的货物处理	31
2.4 系统集成应用的考虑	32
2.4.1 系统集成的过程中到底要完成什么任务	32
2.4.2 如何保证系统集成过程中数据传递的可靠性	33
2.4.3 如何使用消息服务器	34

2.5 实战: PLM 数据与现有系统的集成	35
2.5.1 项目背景	35
2.5.2 业务痛点	36
2.5.3 技术难点	37
2.5.4 解决方案及经验教训	40
2.6 总结	42
第3章 系统之间相互作用的模式	43
3.1 系统集成模式简介	43
3.2 系统集成模式中几个最重要的概念	48
3.2.1 主题与队列在消息传递中的区别	49
3.2.2 消息服务器所使用的储存-转送	50
3.2.3 消息服务器的容错和高可用性	53
3.2.4 分级式事件驱动架构及其实际应用	55
3.3 系统集成模式的实战应用和分析	58
3.3.1 消息的顺序处理	58
3.3.2 持久订阅如何实现	61
3.3.3 命令类消息的应用	64
3.3.4 事件消息的使用	66
3.3.5 回复地址的使用	67
3.3.6 消息传递搭桥的使用	68
3.3.7 消息信封的使用	72
3.4 总结	73
第4章 常见的参与集成的功能系统	74
4.1 功能系统与集成基础设施的连接	75
4.2 常见功能系统的功能和类型	77
4.3 总结	78
第5章 究竟什么是服务	79
5.1 什么是服务	79
5.2 是谁在推动服务的重复使用	81

5.3	服务的操作	82
5.4	服务的界面	85
5.5	服务操作的粒度	89
5.6	服务的组合——SOA	91
5.7	实战：数据	93
5.7.1	项目背景	93
5.7.2	业务痛点	93
5.7.3	技术难点	93
5.7.4	解决方案及经验教训	94
5.8	总结	95
第6章	系统集成项目的实施步骤	97
6.1	系统集成与服务项目概述	98
6.2	系统集成与服务项目的具体实施步骤	101
6.3	设计和开发阶段	102
6.3.1	搜集项目业务功能要求	102
6.3.2	架构设计	107
6.3.3	细节设计	110
6.3.4	代码编写和单元测试	112
6.3.5	集成测试	114
6.4	测试和验收阶段	114
6.4.1	质量保证(QA)部署	115
6.4.2	质量保证(QA)测试	116
6.4.3	用户验收(UA)部署	116
6.4.4	用户验收测试(UAT)	116
6.4.5	(可选项)操作验收测试(OAT)	117
6.5	运维、培训和交付阶段	117
6.5.1	生产环境部署	117
6.5.2	试运行	117
6.5.3	培训及文档提交	118
6.5.4	项目验收	118
6.6	总结	118

第7章 集成项目与公共服务	119
7.1 公共服务的具体内容	120
7.1.1 日志服务	121
7.1.2 出错处理服务	124
7.1.3 ID映射服务	126
7.1.4 顺序处理服务	128
7.1.5 系统及应用监控服务	129
7.1.6 应用、服务、API的分析服务	131
7.2 业务项目的项目模板及其与公共服务的互动	132
7.3 总结	133
第8章 SOA 在实施中的局限性	135
8.1 SOA 在具体实施中的做法	136
8.1.1 SOA 的设计原则	136
8.1.2 SOA 绩优中心	139
8.2 深挖 SOA 的初衷	142
8.3 SOA 的适用范围和局限性	144
8.4 总结	145

第2部分 正篇——现代API、应用互联网

第9章 现代API的引入、应用互联网	149
9.1 什么是(现代)API	149
9.1.1 REST架构的特点	151
9.1.2 REST架构的特点在API中的具体应用	153
9.2 (现代)API流行背后的原因	157
9.2.1 API和云平台的普及	157
9.2.2 API与企业数字化转型、应用互联网以及API经济	158
9.3 API的平台和工具有待进一步地统一和标准化	159
9.4 一个REST API的结构	162
9.5 对API的认识不是一蹴而就的	166
9.6 动手开发API——先尝为快	168
9.7 总结	175

第 10 章 围绕 API 的开发工作	177
10.1 API 的生命周期	178
10.1.1 API 的设计生命周期	178
10.1.2 API 的运维生命周期	184
10.2 API 的调用者	195
10.3 API 项目中的人员和流程	198
10.3.1 什么是使能中心	200
10.3.2 围绕使能中心的不同角色	200
10.3.3 使能中心与绩优中心的区别	201
10.3.4 建立使能中心的具体步骤	202
10.3.5 建立使能中心的好处	208
10.4 总结	209
第 11 章 API 与微服务	211
11.1 什么是微服务	211
11.2 微服务与服务的关系	213
11.3 微服务与 API 的关系	215
11.4 总结	216
第 12 章 API 与云计算	217
12.1 云计算需求的由来	217
12.2 云计算对 API 技术的影响	219
12.2.1 云计算的平台能为你的 API 和应用提供多少服务	219
12.2.2 现有系统之间的连接是否受到影响	221
12.2.3 是否需要增加安全措施	222
12.2.4 如何将 API 负责对内和对外的部分分开	224
12.3 实战：全云和云-本地混合型的 API 平台	228
12.3.1 项目 1 背景	229
12.3.2 项目 1 云平台的架构	229
12.3.3 项目 2 背景	230
12.3.4 项目 2 混合型平台的架构	230
12.4 总结	231

第 13 章 最佳实践的经验	233
13.1 关于系统集成的最佳实践	235
13.1.1 不要以“数据复制”的思考方式设计系统集成	235
13.1.2 尽量避免使用批处理文件的方式	236
13.1.3 对消息服务器运行的认识	237
13.1.4 使用 SEDA 的架构模式来提高系统集成整体设计的可靠性	238
13.1.5 对容错、负载平衡和高可用性的考虑	240
13.1.6 对灾难恢复设置的考虑	242
13.1.7 接收 JMS 消息时的消息确认方式对消息处理可靠性的影响	243
13.2 关于 API 的最佳实践	244
13.2.1 在设计 API 的过程中使用“资源”的字眼,而不要使用“数据”	244
13.2.2 不要使用 API 的概念和方式来做系统集成	245
13.2.3 API 还是连接器	246
13.2.4 API 实施中的出错处理	249
13.2.5 API 的 URI 的每一个部分都应该是名词,而不是动词	249
13.2.6 API 的版本管理	251
13.3 关于架构设计的最佳实践	254
13.3.1 不要使用 UML 的时序图来编写系统集成的用例文件	254
13.3.2 注意区分设计中功能方面和非功能方面的要求	255
13.3.3 不要在没有系统性能指标要求的情况下对系统进行性能的评价和测试	255
13.3.4 数据验证逻辑与数据的关系	256
13.3.5 API、服务和集成中均不保留状态	256
13.4 总结	257
第 14 章 围绕 API 的展望	259
14.1 关于企业的 IT 欠债	260