

# 颠覆市场而非 IT 团队；简单 7 步， 一次就让您的 API 架构回到正轨

作者：Ian Murphy，Creative Intellect Consulting 首席分析师



软件正在吞噬整个世界。这种观点在很多大型企业中越来越流行，事实也确实如此。看看我们的周围，每天都有几百个新的移动应用问世。其中不只有休闲游戏类应用，很多应用瞄准的还是消费者和企业对企业 (B2B) 群体。移动应用的爆炸式发展源于软件构建和设计方法经历的一系列转变。

许多企业、软件或系统架构师想要利用软件获得差异化竞争优势，但是却不知道从何下手。其实关键在于通过应用编程接口 (API) 向合作伙伴、客户和第三方开放内部系统，这样，开发人员就能更轻松地实现系统互联。

听起来很简单。但是事实上，向一堆蜂拥而至的人开放核心系统可不像开门那么简单。幸运的是，架构师可以利用很多方法来构建一个成功的 API 架构。

## API 的优势：速度更快，费用更少，集成一体

如果处理得当，开发和公开 API 可以让企业受益匪浅。借此，企业能够加速实现与业务合作伙伴的集成，从而提高自动化水平，降低成本。此外，API 还将吸引新的合作伙伴，这些新加入的合作伙伴又能帮助企业进军新市场。在企业内部，API 也能减少 IT 部门因推出新的竞争性解决方案而产生的巨额财政赤字。通过加快应用开发速度，API 能够减少因业务部门寻求企业外第三方开发应用而产生的风险。如果业务部门依然坚持使用第三方应用，API 也能确保这些第三方能够将应用妥善集成至企业系统中。

从架构的角度来看，API 在简化、集成和加速新应用的开发方面极具价值。但是，为了充分利用 API 战略，您还需要考虑一些因素。Creative Intellect Consulting 报告 [《交付成功 API 战略的关键因素》 \(Critical factors in delivering a successful API strategy\)](#) 详细讨论了您需要考虑的重要因素。在本文中，我们总结成了 7 步，希望为架构 API 之旅的规划人员提供一些启示。

## 1

# 在行动之前先提出大问题

“我们为什么要这样做”和“谁会使用 API”，这可能是任何架构师都需要提出的最大问题。不理解构建和公开 API 的目的，也就不可能开发出安全有效的 API。

答案可能很简单，比如，为了构建第三方开发人员生态系统，加强与客户的互动。很多银行、零售商和旅游公司的情况确实如此。借助 API，第三方为这些企业开发了许多应用。这些第三方应用不仅让客户受益，还增强了企业与客户互动的能力。

其他企业的关注点可能是如何推进数字化转型。借助应用，用户能在更短的时间内完成任务。比如，通过支付应用，用户能访问多个支付系统，付款给多家供应商。也就是说，付款时，客户不用大费周章地跑去银行填写付款单。整个流程变得更快了，转账的时间也缩短了。同时，支付应用还能帮助用户、银行和收款人节省成本。

## 2

# 适可而止：选择您想公开的 API 代码

为了连接不同的系统，IT 部门使用了一个庞大的内部代码库。选择公布哪些代码作为 API，即使是向内部开发人员公开，都是一件需要深思熟虑的事情。以前也有公司尝试公开企业系统，但是通常都以失败告终。究其原因是因为他们没有正确地划分能够公开和不能公开的代码。比如，他们只是草草地拿出手上一大段连接订单系统的代码，并且毫无保留地公诸于众。但结果就是企业面临安全问题，亦无法便捷地使用和有效地管理由此生成的 API。

如果用户需要的只是查询库存水平和下单，那么开发人员可以快速确定代码数量，而 API 就只公开这些功能。这一点同样适用于单个 API。如果开发人员需要扩展应用以增加付款选项，那么他们不妨将这个选项做成一个单独的 API。相比一个大型 API，两个小型 API 的维护、版本控制和稳定性的维持无疑要更加简单。

在做决定时，请务必回答以下问题：我最少需要公开多少 API 代码？这种做法有助于确保 API 有焦点、有目标。而 API 的用户也会知道他们将获得的功能，并充分利用 API。就更不要提您还能更轻松地了解小型 API 的安全性了。

务必先与 API 用户交流，再开始决定应该在 API 中公开哪些代码。您需要了解他们的真正需求。他们的要求和需求可能有极大的差异。这个过程有助于您确定是以一个 API 还是多个 API 公开这些代码。同时，您还能从这种交流中，知道用户对 API 功能的期望。如果您能先验证 API 需求再构建 API，那么 API 的成功几率将大大提高。此外，因为只开发了必要的 API，这无疑又减少了 API 开发工作。

## 3

# 仍然需要中间件

我们很容易就可以想到，公开 API 后，开发人员就能够针对核心系统编写代码。当然这本身就是很多 API 的目的，但是直接访问却成了一个安全问题和系统问题。如果 API 的用途是支持用户访问企业数据库用于内部分析，那么很有可能会出现这样一种情况：几个名字不同的数据库中有些字段的数据可能一模一样。为了便于用户使用，我们有必要提取 API 和后端系统之间的联系。

这种提取（即，中间件）有很大的好处。这意味着，后端系统的变化不会影响 API。中间件层接受一端应用发出的请求，然后映射至另一端。比如，当后端数据库中的某个字段名出现变动时，只有与该字段映射的中间件会出现相应的变动，所有与该数据库交互的其他应用将继续保持正常运行。这种方式有助于确保系统的稳定性和可靠性。

同样的，API 的变动和版本变更也不会影响后端系统。因此，任何利用 API 开发的应用也将保持稳定和可靠的状态。任何一端出现变动时，只有提取的中间件会随之变动。

## 4

### 架构设计过度导致速度受到影响

在构建 API 时，切记架构设计不要过度，亦不要将 API 复杂化。举一个例子，付款流程所用的 API 需要符合多个层级的合规性。付款流程会捕获信用卡、借记卡或用户的银行详细信息，而这些数据的捕获、加密、使用和存储都有严格的规定。API 必须经过严格的测试，以确保任何使用该 API 的应用都能正确地处理这些数据。同时，这也代表您必须谨慎地安排和规划任何更新，以确保这些更新都经过了严格的测试。

如果 API 为用户提供的是有关天气或交通状况的信息，那么基础合规性的级别又截然不同。在这种情况下，API 并未捕获任何个人信息，只是根据用户的选择标准，反馈从多个数据源捕获的数据。这时自然不存在合规性问题，数据管理的测试级别也就更低。因此，您也能更快速地交付 API，并更频繁地更新 API。

## 5

### 跟踪订阅和使用流程

开发人员需要订阅才能使用 API。这个流程对于内部用户和外部用户（比如，第三方开发人员）都是一样的。通过该流程，API 的所有者和使用者之间就建立了一种关系。当 API 的所有者更新或变更 API，亦或更新 API 版本时，他们只需要简单地发一封电子邮件给订阅者通知他们 API 的变更信息即可。

借助订阅流程，API 的使用者和所有者之间能展开有效的对话。借此，您可以调查用户对新功能的需求。具体来说，您可以在用户群中针对新功能发起投票，以便确定开发人员的时间，确保 API 始终保持与时俱进并能满足用户的需求。同时，这还有助于培育 API 社区，鼓励开发人员基于 API 开发新工具和实用程序。

## 6

### 克服代码膨胀问题：删除过时代码，这种做法永远都不会为时过晚

不论是企业内部编写的代码还是商务软件包的代码，很多代码都有一个问题：代码膨胀。尽管有些膨胀是因为不加选择地增加新功能所致，但是大多数膨胀还是因为增加新代码时没有删除过时的旧代码（通常是没有使用的代码）。由于没有一个流程来跟踪代码的最新使用日期，导致企业很少移除旧代码。随着 API 规模的增大，它需要的资源更多，留下旧代码的风险也就越大。

我们可以采用多种途径解决这一问题。Salesforce 公司的做法是，他们将支持当前版本的平台和 API。新版本推出后，将会给客户几个月的时间来评估和解决现有的软件。所有应用都将在某个时候迁移至最新版本的平台上，任何没有解决的 API 变更都将是最终用户的责任。

其他企业则预备支持旗下产品的多个版本。Red Hat 公司今日宣布，他们将支持每个版本的软件，最长可达 10 年，旨在为客户提供更长的软件使用寿命，并确保软件的稳定性。但是这并不表示他们不更新或刷新其 API，只是说他们将继续支持基础软件。

至于支持 API 或 API 内的功能多久，则由每个企业自行决定。企业支持的版本越多，投入的支持资源也就越多，并且这还需要长期累积的技术知识。企业支持的版本越少，他们与 API 用户发生冲突的风险就越高。

## 7

### 安全性：为什么越小越安全？

您很难发现和解决潜在的安全问题，这是大型 API 的一大风险。如果能维持一个小规模的 API，当出现安全问题时，您就能快速测试 API，然后快速更新和发布 API。这种方法比代码库更可靠。API 的订阅机制又能让提供商和使用者都知道使用 API 的地方以及基于该 API 开发的应用类型。

这种方法的主要优势在于风险管理。当您知道在哪些地方使用了 API，您就可以查看更新，决定是立即更新 API 还是在更“空闲”的时段更新 API。而通过传统的代码库，大型企业通常不可能发现哪些地方使用了 API。结果导致风险评估不足，也没有更新关键的安全补丁。

## 提出正确的问题，作出正确的解答

对于企业来说，实施由 API 驱动的环境并非可有可无，而是一项能使企业长期受益的工作。在启动这项工作之前，架构师必须先提几个简单的问题，比如：“我们为什么要构建这个 API”，以及“它能给我们带来哪些业务优势”。通过验证 API 潜在用户的要求，您可以维持小规模 API，并专注于满足最终用户的需求。订阅流程意味着您在进行版本控制和采取安全措施时，所有用户都将得到通知，并执行相应的更新。

API 有可能会彻底改变软件编写的方式。利用 API，企业能支持第三针对核心后端系统编写应用，同时不给企业带来风险。架构师负责高效地设计企业的 IT 环境。通过在 API 上投入时间，您能够打造一个平台来加快交付新应用，从而帮助您的企业和合作伙伴开发有竞争力的系统。

如欲了解 IBM 如何实施 API 驱动型环境的更多信息，请点击此处：

[ibm.biz/BdsFXK](https://ibm.biz/BdsFXK)