

目 录

致谢

开源项目挣钱实用手册

致谢

当前文档《开源项目挣钱实用手册》由 进击的皇虫 使用 书栈(BookStack.CN) 进行构建，生成于 2018-03-29。

书栈(BookStack.CN) 仅提供文档编写、整理、归类等功能，以及对文档内容的生成和导出工具。

文档内容由网友们编写和整理，书栈(BookStack.CN) 难以确认文档内容知识点是否错漏。如果您在阅读文档获取知识的时候，发现文档内容有不恰当的地方，请向我们反馈，让我们共同携手，将知识准确、高效且有效地传递给每一个人。

同时，如果您在日常生活、工作和学习中遇到有价值有营养的知识文档，欢迎分享到 书栈(BookStack.CN)，为知识的传承献上您的一份力量！

如果当前文档生成时间太久，请到 书栈(BookStack.CN) 获取最新的文档，以跟上知识更新换代的步伐。

文档地址：<http://www.bookstack.cn/books/FinancialSupportForOpenSource>

书栈官网：<http://www.bookstack.cn>

书栈开源：<https://github.com/TruthHun>

分享，让知识传承更久远！感谢知识的创造者，感谢知识的分享者，也感谢每一位阅读到此处的读者，因为我们都将成为知识的传承者。

开源项目挣钱实用手册

- [开源项目挣钱实用手册](#)
- [目录](#)
 - [捐赠按钮](#)
 - [优点](#)
 - [缺点](#)
 - [案例学习](#)
- [悬赏](#)
 - [优点](#)
 - [缺点](#)
 - [案例学习](#)
- [众筹（一次性）](#)
 - [优点](#)
 - [缺点](#)
 - [案例学习](#)
- [众筹（持续性）](#)
 - [优点](#)
 - [缺点](#)
 - [案例学习](#)
- [卖书及周边](#)
 - [优点](#)
 - [缺点](#)
 - [案例学习](#)
- [广告](#)
 - [优势](#)
 - [缺点](#)
 - [案例学习](#)
- [受雇于公司并继续你的项目](#)
 - [优点](#)
 - [缺点](#)
 - [案例学习](#)
- [在职时启动项目](#)
 - [优点](#)
 - [缺点](#)
 - [案例学习](#)
- [补贴](#)
 - [优点](#)
 - [缺点](#)
 - [案例学习](#)
- [咨询服务](#)

- 优点
 - 缺点
 - 案例学习
- SaaS
 - 优点
 - 缺点
 - 案例学习
- 双重协议
 - 优点
 - 缺点
 - 案例学习
- 开放核心
 - 优点
 - 缺点
 - 案例学习
- 基金会
 - 优点
 - 缺点
 - 案例学习
- 风险投资
 - 优点
 - 缺点
 - 案例学习
- 贡献
- 协议

开源项目挣钱实用手册

“我在参与开源项目，但是我该如何挣钱呢？”

我列出了我从各种渠道所知道的开源项目带来收入的人们故事，每种出资类型都有一系列的真实案例，粗略的按照出资金量从小到大排列。（我已尽可能的将链接指向具体故事而非主页面）

本文中的出资类型并非互斥的，例如一个项目既可以由基金会也可以通过众筹来筹集资金，而一个人既可以靠咨询挣钱也可以获得捐赠。本文的主要目的是提供一个详尽的挣钱方式列表，而你只需要从中选出适合你的。

本中文版是[原版](#)的翻译版本。

原项目名称Lemonade Stand是指销售柠檬汁的小摊，而在美国，这种小摊通常是由小朋友运作的。

目录

1. [捐赠按钮](#)
2. [悬赏](#)
3. [众筹（一次性）](#)
4. [众筹（持续性）](#)
5. [卖书及周边](#)
6. [广告](#)
7. [受雇于公司并继续你的项目](#)
8. [在职时启动项目](#)
9. [补贴](#)
0. [咨询服务](#)
1. [SaaS](#)
2. [双重协议](#)
3. [开放核心](#)
4. [基金会](#)
5. [风险投资](#)

附录：[贡献](#) // [协议](#)

“个人努力”用来标记其资金是由个人而非项目主导筹集获得的

捐赠按钮

在你的网站页面里放上捐赠按钮。Stripe和PayPal都可以很方便的提供这项服务。

优点

- 限制条件少
- 工作量小：放好后就可以不管了

缺点

- 除非你努力筹款，通常都不会有太多钱
- 需要一个法人实体来接受捐赠（[SFC](#)，[OpenCollective](#)可在这方面提供帮助），因此较难管理个人的国际性捐赠
- 在多人项目中很难明确如何分配这笔捐赠

案例学习

- [Twisted](#)
- [Git](#)

- [Transmission](#)

悬赏

项目或公司可能时不时的张贴开源项目的悬赏工作（例如“修漏洞赚\$100”），下面列出了一些网站，他们收集和发布了这类悬赏工作。

优点

- 参与到开源社区
- 确定的工作有确定的回报
- 在修复安全漏洞方面特别流行

缺点

- 会在项目中产生不合理的激励机制（低质量的PR也会影响项目的专注性）
- 通常没多少钱（~<\$500）
- 无法提供持久的收入

案例学习

- [Bountysource](#)
- [GitHub Bug Bounty Program](#)

众筹（一次性）

如果你想实现一个特别的想法（区别于长期项目），一次性的众筹活动可以帮助你筹集到你需要的资金，许多个人和公司都可能会为你的想法捐款。

优点

- 限制条件少
- 对个人来说也可以很容易合法的进行，例如通过[Kickstarter](#)

缺点

- 一大堆的市场工作需要做
- 通常都需要为捐赠者提供回报甚至是特权
- 通常钱也不是那么多（一次大约\$50K）
- 公司并不常愿意向众筹捐款

案例学习

- [Michal Papis + Rvm](#) (个人努力)
- [Andrew Godwin + Django](#) (个人努力)
- [ribasushi + CPAN](#) (个人努力)
- [RESTful WP-CLI](#)

众筹（持续性）

如果你想要为持续性的项目筹集资金，可以设立一个持续性的众筹，捐赠者承诺按月或按年提供资金直到捐赠者退出。对于那些经常使用你的项目的个人或公司可能会愿意为你的项目提供资金。

优点

- 限制条件少
- 对个人来说也可以很容易合法的进行，例如通过[Patreon](#), [Salt](#), [Gratipay](#), [OpenCollective](#)

缺点

- 很难获得承诺的持续性捐赠（这通常需要你或项目已经有一定的名声）
- 很难解释持续性的捐赠能获得什么样明确的回报或特权
- 通常钱也不是那么多（一个月\$1-4K）

案例学习

- [MochaJS](#)
- [React-boilerplate](#)
- [jsbin](#)
- [Tom Christie + Django REST framework \(personal effort\)](#)
- [Ruby Together](#)

卖书及周边

如果你是某个领域的专家，你可以写书卖书，可以找个出版社（像O'Reilly）或自己出版（译者注：在中国不行）。除了卖书之外，有些项目也卖短袖外套等。

优点

- 筹集到的资金并不和项目本身管理关联，所以项目本身可以保持创作的自由
- 销售的商品本身也可以当做项目做的宣传

- 在初次销售后可以成为持续性的资金来源

缺点

- 通常钱也不是那么多
- 会影响用在项目上的精力
- 卖周边需要准备预付资金

案例学习

- [Lua](#)
- [Daniel and Audrey Roy Greenfeld + Two Scoops of Django](#) (个人努力)
- [Sandi Metz + Practical Object-Oriented Design in Ruby](#) (个人努力)

广告

如果你的项目已经有了一定的受众，你可以帮助广告商向你的受众推销。通常你的项目都会有明确的受众，这是你的优势也是广告商所喜欢的。（比如你有一个Python项目，那么基本上可以假定你的受众一定是技术上熟悉Python的）

优势

- 这是成熟明确而且大众也能接受的商业模式

缺点

- 需要足够多的受众才能请来广告商
- 需要通过透明化来让受众相信你（比如让其信任你不会追踪他们）
- 需要许多精力来寻找和管理广告商

案例学习

- [Read the Docs](#)
- [Hoodie](#)

受雇于公司并继续你的项目

公司有时候会雇佣一些个人来做开源项目，你可以寻找一个正在使用你的开源项目的公司。当然具体在公司里可能公司工作和开源项目工作时间会是五五分。除此之外，也可以找一个愿意尝试使用你的新项目的公司。如果你有展示项目经验，这将会非常有用。

优点

- 可以利用公司的资源
- 可以很好的和公司的需求保持一致
- 稳定的收入

缺点

- 获得这样的机会需要极好的运气，现目前没有明确可重复的方式获得这样的机会
- 项目通常需要非常出名并且被使用
- 对于没有为公司的利润工作，这使得个人很容易被公司优先舍弃
- 公司可能会过分影响项目的发展
- 可能会因平衡不好两边而影响项目

案例学习

- [Donald Stufft + Hewlett-Packard and Python packaging](#) (个人努力)
- [Rich Hickey + Cognitect and Clojure](#)
- [Aaron Patterson + ManageIQ and Ruby, Rails](#) (个人努力)
- [Ryan Dahl + Joyent and Node.js \(opens a YouTube video\)](#) (个人努力)

在职时启动项目

许多开源项目最初都是员工的编外项目(Side Project)，即便其最终可能会成长为一家公司，但以编外项目的形式在公司里进行孵化应该是不错的选择。

如果你想走这条路，请确定你理解了公司在开源项目上的政策。有些公司鼓励员工在工作时间从事开源项目开发，而有些则将你的任何工作视作公司项目。不要假定任何前提，最好在开始前问问你公司里的相关人员。

优点

- 可以不用担心收入的情况下尝试新想法
- 可以和公司的需求很好的保持一致
- 适合尝试新想法

缺点

- 需要用业余时间开发，或者获准在工作时间开发
- 有被公司过分影响的风险
- 持续下去可能会出现极度复杂的管理情况

案例学习

- [Mozilla and Rust](#)
- [Google and Go](#)
- [Facebook and React](#)
- [Futurice's open source program](#)

补贴

补贴是不需要偿还的极其有效的大笔捐赠，提供补贴的组织通常能够通过给予补贴而从其他方面获得利益，例如接近你，展示其影响力，获得你的工作汇报或税率优惠。

补贴可能来自很多地方，包括公司、软件基金会、慈善基金会以及政府，其技术及法律方面会因其来源的不同有很大的差异。比如一家公司可以通过开咨询费发票给你补贴，而慈善基金会则只能给非盈利组织或个人，通常你得找到一个非盈利组织来帮助你。如果你对补贴不熟悉，了解它的最好方式就是和曾经获得过的人去了解。下面列出了一些成功的案例

优点

- 限制条件少
- 有保证的资金可以确保你能在一段时间里专注在你的项目上
- 让你的项目有时间喘口气和做些试验

缺点

- 软件方面没太多提供补贴的组织
- 补贴是有限的，始终需要在用完补贴前找到一个持续方法

案例学习

- [Dat](#)
- [@shazow/urllib3-stripe-and-open-source-grants-edb9c0e46e82#.45ylnxrh4">Andrey Petrov + Stripe Open-Source Retreat and urllib3](#)
- [Django + Mozilla Open Source Support](#)

咨询服务

咨询是一种为开源项目提供资金的灵活方式。你可以更加自由的规划你的时间，比如一周30小时做咨询业务，10小时做开源项目。咨询师通常收费相对较贵，因为工作不稳定且没有公司福利。如果你打算做这类公司，你应该想要创建一家有限责任公司。

优点

- 这是成熟明确而且大众也能接受的商业模式

缺点

- 咨询工作需要人力，而且不宜规模化（除了极少数）
- 商业本身的需求会分散开源项目上的注意力
- 可能会和软件的简易要求有差异
- 项目需要足够流行，才会让人愿意为相关服务付钱

案例学习

- [Neighbourhoodie](#)
- [Baroque Software](#)
- [OpenSSL](#)

SaaS

SaaS 即 [软件即服务\(Software as a Service\)](#)。在这个模型下，代码本身是开源的，但是你可以提供增值服务使得用户更容易使用。一个典型的增值服务就是托管付费。

优点

- 可以围绕这个开源项目建立一个社区，然后通过托管服务赚钱
- 让开源项目可以专注在用户层面，当需求增加后再帮助企业采用这个项目
- 可以根据用户数进行规模化改造

缺点

- 对增值服务使用者来说，通常意味着托管服务必须比招聘一个人来维护项目更便宜
- 增值服务有可能会使得免费用户不太高兴

案例学习

- [WordPress.com](#)
- [Moodle](#)
- [Forge Laravel](#)
- [Gitlab](#)
- [Sentry](#)
- [Travis CI](#)
- [Ghost](#)

双重协议

有时候项目可以在完全相同的代码里提供两种不同的授权协议：一个是商业友好的，而另外一个则不（例如GPL）。后者对于个人来说可以免费使用，而公司需要通过购买商业协议来获得合法的商业授权。

优点

- 这是成熟明确而且大众也能接受的商业模式
- 如果成功的话，也可以做到规模化

缺点

- 会和让软件自由获得的理想有冲突
- 项目需要足够大以满足客户的需求

案例学习

- [MySQL](#)
- [SQLite](#)

开放核心

在[开放核心](#)模型中，项目的一些方面是免费的，但一些功能则是私有的，且只对付费用户开放，通常要求这个项目有企业的需求。

优点

- 这是成熟明确而且大众也能接受的商业模式
- 如果成功的话，也可以做到规模化

缺点

- 需要设计付费项目，且该项目应该是独有的
- 会和让软件自由获得的理想有冲突
- 独有功能有可能会使得免费用户不太高兴

案例学习

- [Docker](#)
- [Elastic](#)
- [Mesosphere](#)

- [Sidekiq](#)

基金会

基金会是可以接收和支出的合法法人实体。鉴于其目的并非获得利润，因此可以更好保持中立地管理项目。在美国，基金会可以是501c3（非盈利）或501c6（贸易联盟），许多软件基金会都是贸易联盟，因为非盈利基金会要求展示出慈善的目的，这在软件开发中比较困难。

优点

- 中立，基金会可以帮助保护代码和管理社区
- 可以在许多捐赠者中散布影响力
- 使得项目合法，公司会更愿意向基金会付款/捐赠而非个人

缺点

- 只适合大项目
- 由于IRS的限制，项目能做什么会有所限制
- 需要大量社区的努力和各种技能，而且之后仍旧需要努力获得筹款

案例学习

- [Ruby Together](#)
- [Python Software Foundation](#)
- [Node.js Foundation](#)

风险投资

风险投资是高增长业务的一种筹资形式，不像银行贷款或者任意一种债务财务形式，风险投资者通过提供资金来占有你业务的一定股份。这种交易不像贷款，如果你的业务挂了你并不需要偿还出资方。当然，如果你成功了，你也需要成倍的返还给你的投资者。

风险投资是高风险高回报的：风投者相比银行对风险更加宽容，当然他们也期待你成功后的巨额回报。如果你打算获得风险投资，你应该建立股份有限公司。如果你对风险投资过程不熟悉，开始的最好方式就是询问成功获得风险投资的人。

优点

- 制度上的支持对成长中的业务有益
- 大量的风投资金蓄势待发

缺点

- 风险投资在投资之初便做好了获得成倍回报后退出的打算，历史证明，由于开源项目的结构特点，风险投资的成功很难。
- 风险投资者可能会因为优先级改变其动机

（译者注：在中国有一类风险投资者故意诱导涉世不深的创业者签订不平等协议，尤其是在创业者热情最浓而又最困难的时候，使其在业务失败时也能全身而退，相应的，创业者会输得很惨，值得小心对待）

案例学习

- [Npm](#)
- [Confluent](#)
- [NodeSource](#)
- [Meteor](#)

—

贡献

我写这个手册主要是为了将我头脑中的知识都整理出来，不过我并没有打算做主要的贡献或改变。优点和缺点大多是从我的观点出发的主观想法。

如果有什么错误（尤其是案例学习），非常欢迎大家的修改。同时，如果发现有什么分类漏掉了，我也非常欢迎大家的修改。

关于中文版的贡献，如果有信息上的错误或遗失，请到原文中提出，如果有中文翻译上的错误，请在这里直接提出。如果你觉得有信息上的错误或遗失，但又各种原因无法用英文表述，担心中文表述不会被采纳，这种情况请还是在原文中提出，我或者其他同时会英文和中文的小伙伴会帮助解决问题的。

本翻译项目使用[翻译辅助工具](#)辅助同步原文修订部分。

协议

原文协议为Creative Commons CC0 1.0

License，即你可以自由的使用，商业或非商业，不过如果你用了，很开心能从你那里听到点什么，在这里找到我[@nayafia](#)，当然这并不是要求必须做的。

中文版协议为Creative Commons Attribution 4.0 International

License，和原文协议差不多了，基本上可以说怎么用都可以，唯一不要修改原作者名字或以原作者

名字声明演绎作品就可以了。

