

## 第 15 章 探索 GitHub 市场

本章学习重点：

- (1) 介绍 GitHub 市场；
- (2) 在市场中查找应用程序；
- (3) 从市场中安装应用。

在第 5 部分的 3 个章节中，通过研究扩展 GitHub、自定义 GitHub 的多种不同方式以及许多扩展工具或与 GitHub 集成，发现 GitHub 市场是与 GitHub 一起使用的工具的很好方法。

### 15.1 GitHub 市场简介

GitHub 市场 (<https://github.com/marketplace>) 中工具和应用可以分为以下几类：聊天、代码质量、代码审查、持续继承、依赖管理、部署、学习、本土化、手机端、监控、shiyi 项目管理、发布、最近添加、安全、支持、测试、实用工具。

GitHub 市场是在 GitHub 上找到适合各种情况的应用程序的好方法。通过在 GitHub 市场购买或安装应用主要有两个好处：易于计费 and 安装、审查过程。

#### 15.1.1 计费方便

因为可以使用 GitHub 付款信息，通过 GitHub 市场对应用程序进行付款和安装是一个简化的流程。所以如果购买五个不同的应用程序与 GitHub 一起使用时，就不必与五个不同的支付提供商打交道。

如果有一个免费的 GitHub 账户，用户可能没有在 GitHub 中设置付款信息。若要设置付款方式，请单击 GitHub.com 右上角的头像，然后单击设置。在对应页面中，单击左侧列表中的“Billing”，然后可以单击右边的“Add payment method”，如图 15-1 所示。

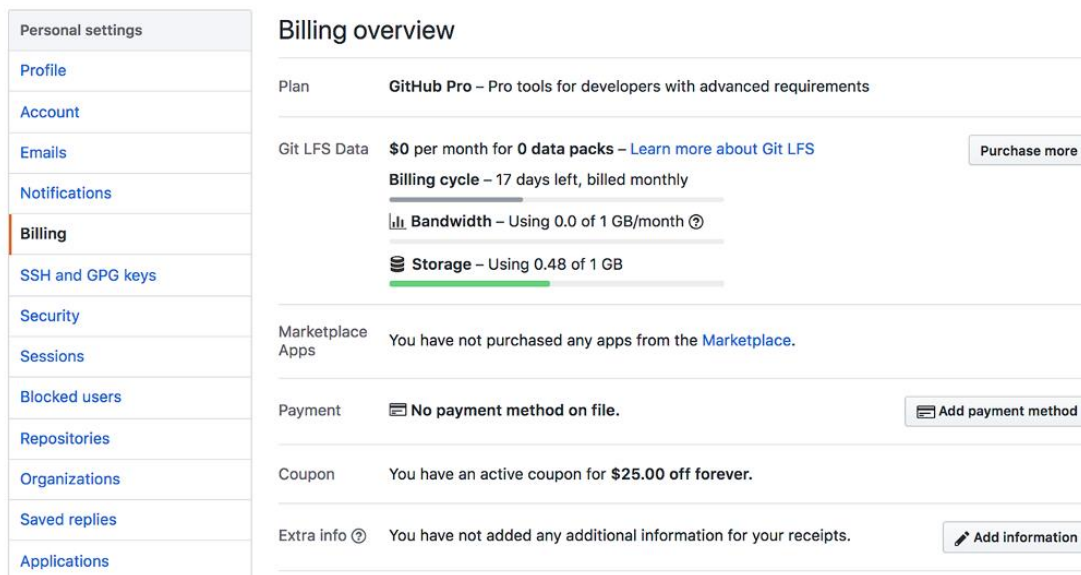


图 15-1 GitHub 的计费方案总览

### 15.1.2 市场审查应用

从市场安装应用程序的一个好处是这些应用程序必须满足特定的要求，GitHub 才会将其列在市场中。这些特定要求有助于确保应用程序具有更高的质量和安全标准；帮助确保这些应用程序有用（不是无用的应用程序）并且是安全的。

**警告：**目前，GitHub Action 不需要在 GitHub 市场中列出任何评论，这意味着安装来自不认识的人的 Action 可能会有点风险。

应用程序必须满足四个主要类别的要求才能在中列出。

（1）用户体验要求：这组简短的九项要求包括应用程序必须拥有一定的用户数量和安装数量，还包括一些围绕应用程序行为的要求。例如应用程序必须包含文档链接，不能主动说服用户远离 GitHub，必须为客户提供价值。

（2）品牌和展示要求：这组指南和建议围绕应用品牌和应用列表。每个应用程序都必须包含自己的徽标。如果应用程序使用 GitHub 的徽标，则必须遵循 GitHub 的徽标和使用指南。页面上的品牌和列表部分包含指向进一步徽标和描述指南的链接。GitHub 非常重视在中列出的应用程序。

（3）安全要求：GitHub 会先对应用程序进行安全审查，然后再将其列入市场。

（4）计费流程要求：市场中的每个应用程序都必须使用 GitHub 市场网络钩子事件来集成计费流程。此要求确保人们可以购买对应用程序的订阅，并使其在 GitHub 上存档的付款信息取消该订阅，还确保通过 GitHub 所做的任何更改都会立即反映在应用程序自己的网站上。

## 15.2 发布应用到市场

在市场中列出自己的应用程序可能会增加应用程序的潜在受众。列出的应用程序需要其满足 GitHub 的要求并获得批准。在第 14 章将指导创建用户自己的应用程序。要开始列出应用程序，请单击市场登录页面底部的“提交工具以供审核”链接或导航到（<https://github.com/marketplace/new>）。此页面列出了可以发布到市场列表的所有应用程序，如图 15-2 所示。请注意，此处列出了在第 14 章中创建的“Why Not Both”应用程序。

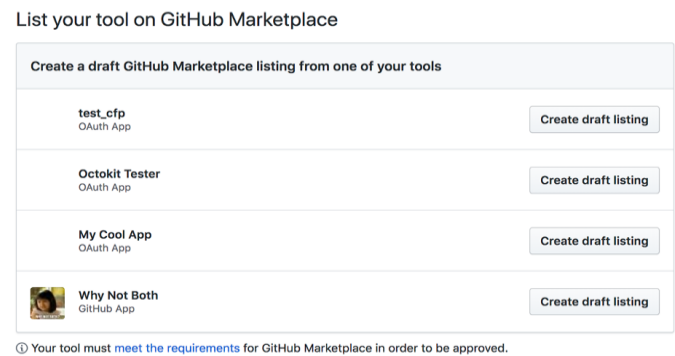


图 15-2 GitHub 市场中的用户应用程序列表示例

单击要在市场上列出的应用旁边的 **Create draft listing** 以开始该过程。这会将带到一个页面，可以在其中输入列表名称，并为本章前面列出的应用选择一个市场类别，如图 15-3 所示。

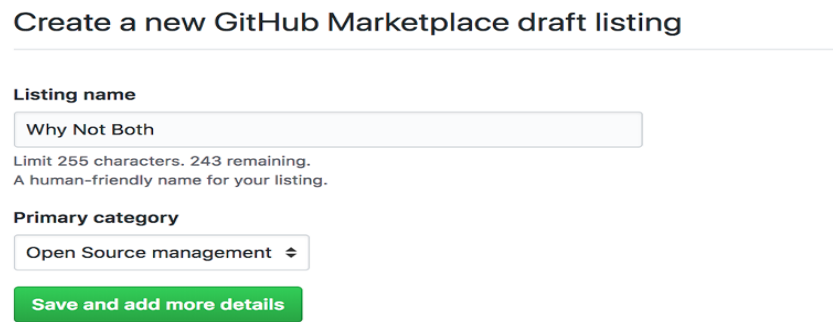


图 15-3 创建新的 GitHub 市场应用程序

提示：如果保存了列表草稿但碰巧关闭了浏览器，则可以通过在浏览器中访问（<https://github.com/marketplace/manage>）来返回到列表。

在填写完这些详细信息后，单击保存并添加更多详细信息以保存列表草稿并继续执行下一组步骤，如图 15-4 所示。

（1）添加联系信息。

此信息是由三个电子邮件地址组成：技术主管、营销主管和财务主管。

（2）填写清单描述

可以在此区域填写更多详细信息，例如产品说明、徽标、和截图。此处的信息将显示在应用程序的市场页面上。

（3）设置定价计划

可以在此处设置一个或多个定价计划，包括选择创建免费计划、月度计划或每个用户月度计划的选项，也可以指定计划是否包括 14 天免费试用。

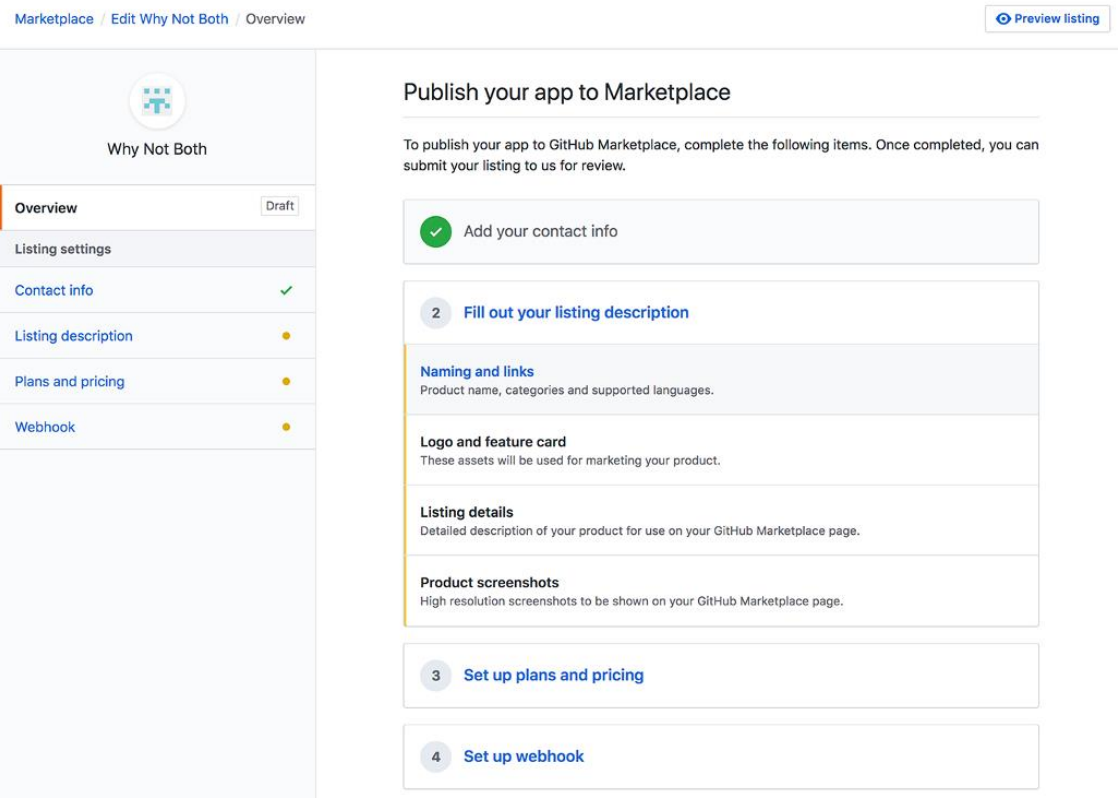


图 15-4 发布应用程序到市场

（4）设置网络钩子

此步骤允许指定一个 URL，其中市场事件将通过 HTTP POST 请求发送到该 URL。网络钩子将向发送有关事件的信息，例如购买、取消以及升级和降级等更改。

（5）接受市场开发者协议

为了在市场上显示应用程序，必须接受市场开发者协议。

（6）单击提交审核按钮

GitHub 员工将审核提交的内容以确保其符合在市场中列出的要求。

15.3 考虑安装常见的应用

在本文开头的“GitHub 市场简介”部分中，已经列出了市场上可用的应用程序类别。本节中会再介绍一些可能想要考虑安装的一些最常见和最有用的应用程序。

### 15.3.1 持续集成

每次将代码推送到 GitHub 时，持续集成 (CI) 应用程序都会自动构建和测试代码。如果仓库上安装了一个 CI 应用，比如 AppVeyor，那么会在每个 PR 的底部看到检查的状态，如图 15-5 所示。

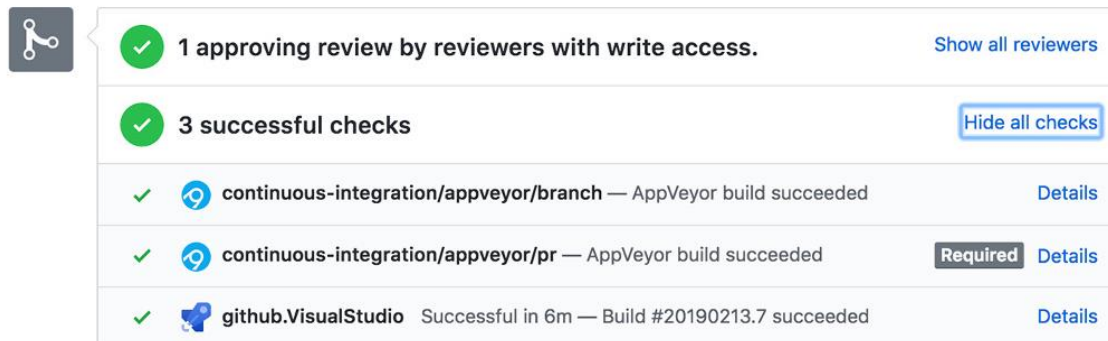


图 15-5 仓库借助持续集成的合并检查

如果是仓库的所有者，还可以指定在分支合并到主分支之前是否必须通过检查。只需进入“Settings tab”选项卡即可。如果已经在主分支上有任何规则，请单击 edit；否则，单击 Add Rule。从那里可以向下滚动并选择在合并前需要 Require 状态检查。

### 15.3.2 代码质量

代码质量应用程序通过样式、质量、安全性和测试覆盖率自动检查代码。这些应用程序对于确保代码保持高标准非常有用。随着使用样式良好且质量好的代码，不太可能引入或遗漏错误。例如，如果要求所有花括号都在新行上，并且每个嵌套花括号缩进一个制表符，那么很可能发现某些不正确的地方。例如，Rubocop 在构建 Ruby 代码时检查其样式，并为用户提供样式反馈，譬如方法名称的大小写等。

另一种有用的代码质量应用程序是代码覆盖应用程序，例如 Codecov。如图 15-6 所示，Codecov 和类似的应用程序会对 PR 进行评论，并说明测试场景覆盖了多少代码，这有助于确保代码保持良好的测试状态。

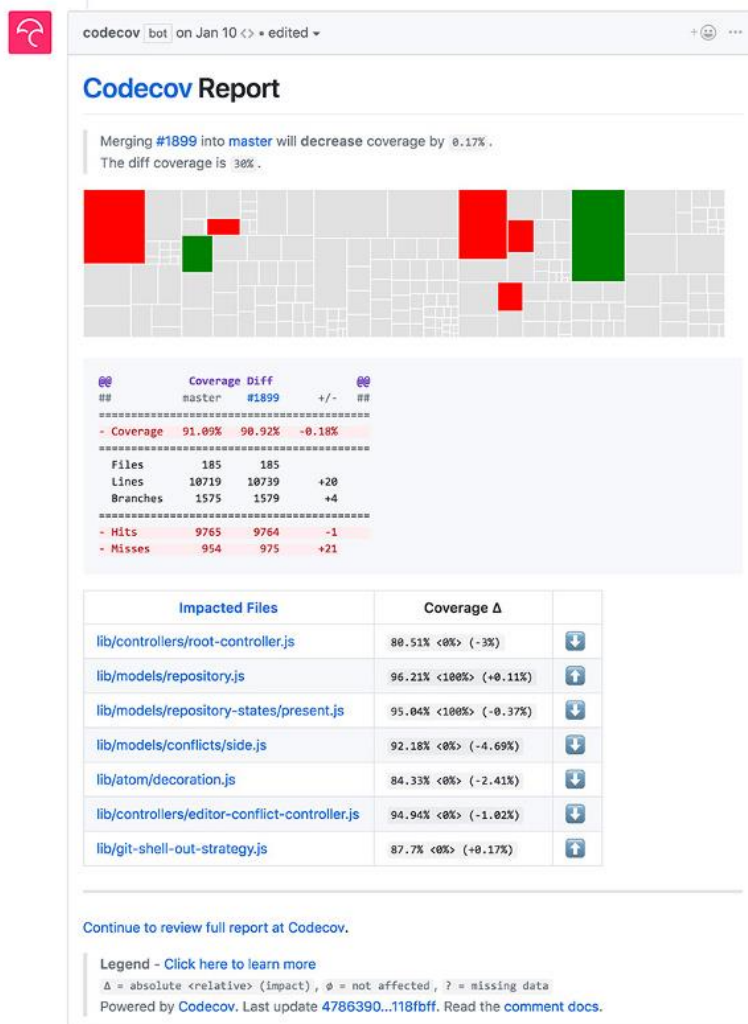


图 15-6 在 PR 中评论的 Codecov 报告

### 15.3.3 本地化

本地化应用程序可以更轻松地以多种语言发布应用程序。例如，Crowdin 应用程序会将仓库链接到 Crowdin 账户，来自世界各地的人可以帮助翻译文档和软件中的任何书面文字（例如，按钮或菜单中的文字）。Crowdin 应用程序有超过 20000 人参与翻译，当达到准确度阈值时，Crowdin 应用程序将自动在仓库中打开一个包含新翻译的拉取请求，仍然让有机会进行审查和合并。对于开源项目，Crowdin 是免费的！

### 15.3.4 监控

监控应用程序有助于衡量性能、跟踪错误和跟踪代码中的依赖关系。例如，Greenkeeper 是一个实时通知应用程序，可为提供 JavaScript 依赖项的更新和更改。图 15-7 显示了 Greenkeeper 的操作，打开 PR 将 eslint 更新到最新版本。



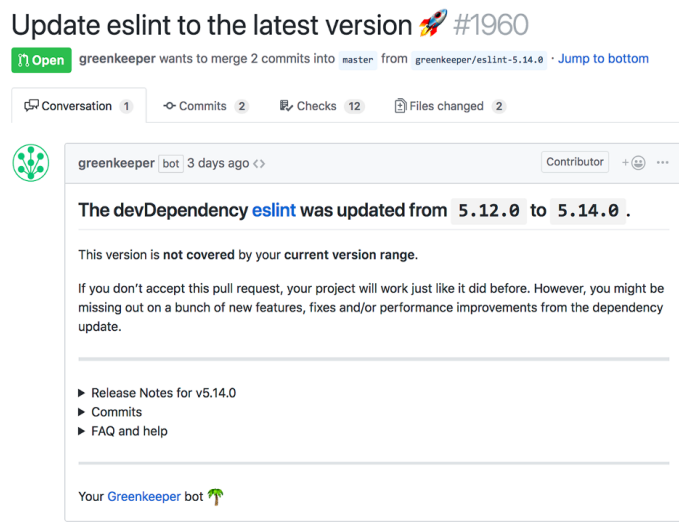


图 15-7 Greenkeeper 打开 PR 将 eslint 更新到最新版本

### 15.3.5 依赖管理

现代的应用程序开发严重依赖公共包管理器来引入和管理依赖项。一个典型的应用程序可能有数十个（如果不是数百个）依赖项。跟踪哪些依赖项是最新的可能很困难，不过像 Dependabot 这样的 Github 应用程序会检查以确保依赖项是最新的，并提交 PR 来更新那些不是最新的依赖项。

有时不希望所有的依赖项都依赖于公共包注册表。例如，如果在一家企业工作，可能拥有应该保持私有的内部包。在这种情况下，私有包注册工具（例如 MyGet）很有用。MyGet 与 NuGet 包配合使用，并允许设置一个策略，其中推送到特定分支将启动构建，并且该分支将部署到托管在 MyGet 上的自定义 NuGet 源。

### 15.3.6 测试

测试软件是软件开发生命周期的重要组成部分。优秀的测试人员会制定测试计划，以确保测试人员能够很好地测试每个版本。管理测试计划和跟踪测试运行状态是质量保证的重要组成部分。TestQuality 应用程序与 GitHub 集成，以帮助开发人员和测试人员创建、运行、协调和监控软件测试任务。

### 15.3.7 学习

提示：学习 GitHub 的一个好方法就是从市场中安装 GitHub 学习实验室。在学习实验室安装一个机器人，通过完成的一组任务，引导完成有关如何使用 GitHub 的交互式课程。该实验室是免费的，可以按照自己的节奏参加任意数量的课程。

## 第 16 章 GitHub 和用户

本章学习重点：

- (1) 充分利用用户个人资料；
- (2) 了解用户贡献图；
- (3) 关注他人并 Star 仓库。

GitHub 通常被描述为开发者的社交网络。虽然 GitHub 远不止是一个社交网络，而是协作开发必不可少的工具集。但是，社交网络仍然是 GitHub 的重要组成部分，而且这很可能是其成功的一个关键原因。在 GitHub 诞生前，就已经有了源代码控制托管平台，但是几乎所有这些托管平台都专注于项目。GitHub 颠覆了这种以项目为中心的理念，而是专注于人。例如，在 URL 中，每个仓库名称前都有用户或组织的名称。

本章将深入研究 GitHub 的社交网络属性，并着眼于用户如何在这个网络中展现自己，以及如何参与在线社区。

### 16.1 了解 GitHub 个人页面

GitHub 上的每个用户都有一个个人页面。图 16-1 显示了本书作者 Phil Haack 的个人页面 <https://github.com/haacked>。

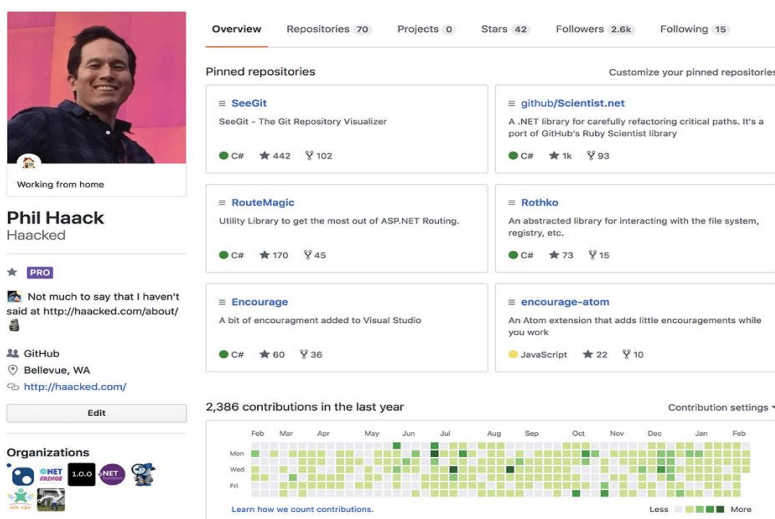


图 16-1 Haacked 的个人页面

个人页面是在 GitHub 上展现自我的地方。当你打开一个 Issue 或第一次在某个仓库提交 PR 时，维护人员可能会查看你的个人页面来了解你。不仅如此，GitHub 个人页面还可以作为开发者的作品集，从而使访问者深入了解开发者的兴趣、经验和能力。许多公司在招聘时都会查看应聘者的 GitHub 个人页面。



警告：许多公司对有 GitHub 个人页面的应聘者给予重视。然而，这对那些不使用 GitHub 工作或者不从事开源工作的人存在偏见。。就连 GitHub 公司自己都不要求有出众的 GitHub 个人页面才能来应聘。越来越多的公司采取这样一种方式：如果应聘者有 GitHub 的个人页面，他们会查看，但如果没有，他们也不会因此抱有偏见。

### 16.1.1 个人头像

个人页面的来访者首先看到的就是用户头像。此图片与用户在 GitHub 上的所有活动相关联。当用户创建一个 Issue、PR 或者留下一个评论的时候，个人头像就显示在旁边。

提示：头像是一个重要的身份标识，用它可以来彰显个性。

### 16.1.2 状态消息

在个人头像图片下方有一条状态消息，状态消息可以向整个社区传递一些用户信息。有的人用它表达一些有趣或有意义的事情，还有的人将它用于实际目的。例如，一个流行项目的维护者想让大家知道因为自己太忙，需要离开 GitHub 一段时间，所以这段时间的消息不能及时回复，他可以点击状态消息进行设置。图 16-2 显示了指定繁忙状态的状态消息对话框。

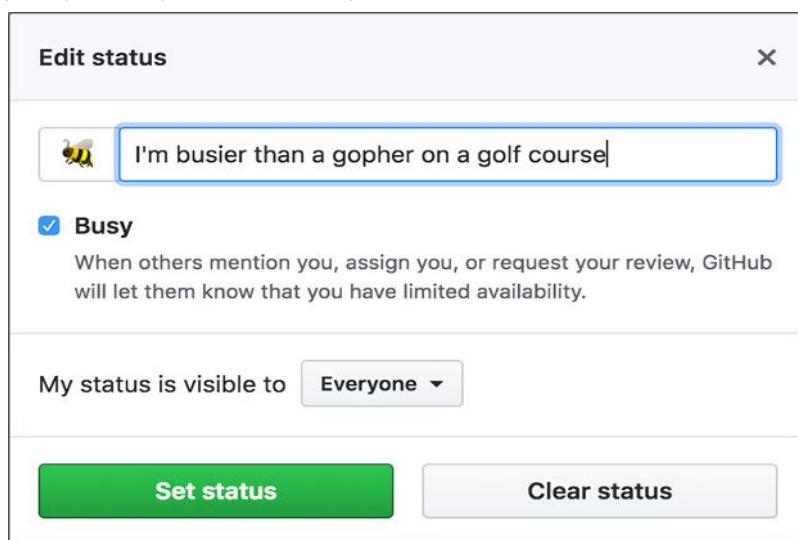


图 16-2 更改你的状态消息

### 16.1.3 个人信息和自述

GitHub 在状态消息下方显示用户公开展示的个人信息和自述。单击“Edit”按钮可以更改个人自述、公司名称、位置和 URL，从而更好地展示并宣传个人博客或个人站点。

注意：所有这些信息都是可选的。输入信息后，在用户个人页面出现的任何地方，GitHub 都有权限显示这些信息。用户可以随时通过编辑个人资料来删除这些信息。

用户有时还会发现 GitHub 添加到用户个人资料中的徽章。例如，在图 16-1 中，你可以看到 Haacked 拥有 Pro 徽章。它表示该用户是订阅了个人专业版的付费用户。在 <https://github.com/pricing> 上可以查看不同的定价模型。Staff 徽章也是一种常见的徽章，专供 GitHub 的全职员工（也称之为 Hubbers）使用。图 16-3 显示了带有 Staff 徽章的个人页面。



图 16-3 个人页面中的“Staff”徽章

#### 16.1.4 固定仓库

默认情况下，GitHub 会在用户的个人页面上显示一些该用户最受欢迎的仓库，但这些仓库对用户来说可能并不是重要的仓库。单击“Customize your pinned repositories”可选择最多 6 个仓库固定到个人页面。

从三个角度来看，固定仓库非常有用：

（1）用户：固定仓库对用户很有用，因为它们为用户最关心的仓库的快速链接。固定仓库可以让用户快速访问地到达那里，而不必为每个仓库创建书签。

（2）其他 GitHub 用户：当其他用户在 GitHub 上访问用户个人页面时，固定仓库是映入眼帘的第一个信息。这会让人们了解该用户对什么感兴趣，以及有经验的领域。如果用户刚刚开始为一个新的开源项目做贡献，那么列出经验领域是很重要的，因为维护人员会经常访问个人页面。

(3) 公司：当用户申请工作时，招聘经理和招聘人员可能会查看你的个人页面，以了解应聘者的经验。把一些贡献最大或最引以为傲的项目放在页面的顶部，可以帮助招聘者对应聘者有一个更准确的印象。

注意：固定仓库可以随时更改。例如，如果用户正在申请一个新职位，固定的内容可能需要更适合所申请的职位。如果用户想学习一些新的东西，固定的内容可能是目前正在使用的仓库。个人页面可以看作是一个动态文档，它可以随着用户的目标和兴趣的变化而更新。

### 16.1.5 贡献图

贡献图是一个高 7 个方格、长 52 个方格的长方形网格，每个方块代表过去一年的某一天。每个方块都填充了一种颜色，代表用户当天的贡献水平。灰色表示没有做出任何贡献，绿色表示做出了贡献，绿色从浅到深表示做出的贡献数量从少到多。

独立仓库中的 Issue 和 PR 会计入贡献。提交到独立仓库的默认分支（通常是 master）或其 GitHub pages 分支（通常是 gh-pages）也会会计入贡献图。

提示：如果用户 fork 一个仓库，那么“Issue”选项卡将不会出现在仓库顶部，因为 fork 一个仓库的目标通常是对原始开源项目作出贡献。用户可以 fork 一个仓库，进行更改，然后向原始仓库发起一个 PR。如果想在 fork 的仓库中拥有“Issue”选项卡，可以在“Settings”选项卡中开启 Issue 选项卡，但通常不建议这样做，因为用户应该跟踪原始仓库中 Issue 的进度。因为 fork 的预期行为是向原始仓库贡献，所以在 fork 的仓库上的 Issue 和不以原始仓库为目标的 PR 将不会会计入贡献图。fork 更像用户自己的私有分支，而真正的贡献是在原始仓库中创建 Issue 或合并代码之后才生效的。

贡献图是 GitHub 比较有争议的特性之一。首先，它会促进不健康的行为，因为许多人试图保持长时间的连续活跃状态。许多人为贡献图中的每一天甚至周末都有活动而感到自豪。但是贡献图的目的不是——也不应该是——鼓励用户在生活中的每一天都写代码。另一个问题是，有的人会根据贡献图对开发人员的能力得出错误的结论。例如，有人可能会查看开发人员的贡献图并看到很少的活动，于是认为他产出不高。

警告：从他人的贡献图中得出的任何结论都是没有意义的。贡献图作为一种查看历史活动的方式，应该只对用户自己有用。事实上，贡献图很容易被操作。有的仓库就可以利用贡献图制作像素艺术（见 <https://github.com/nikhilweee/github-activity-art>）。使用该工具很容易使用户的贡献图完全变成深绿色。因此，不要把贡献图看做一个权威的生产力度量指标。

此外，对私有仓库的贡献可能不会显示在贡献图中。单击“Contributing settings”以更改该设置。图 16-4 显示了启用显示私有贡献和活动图的示例。贡献图显示活动的数量，活动图显示活动的分布。

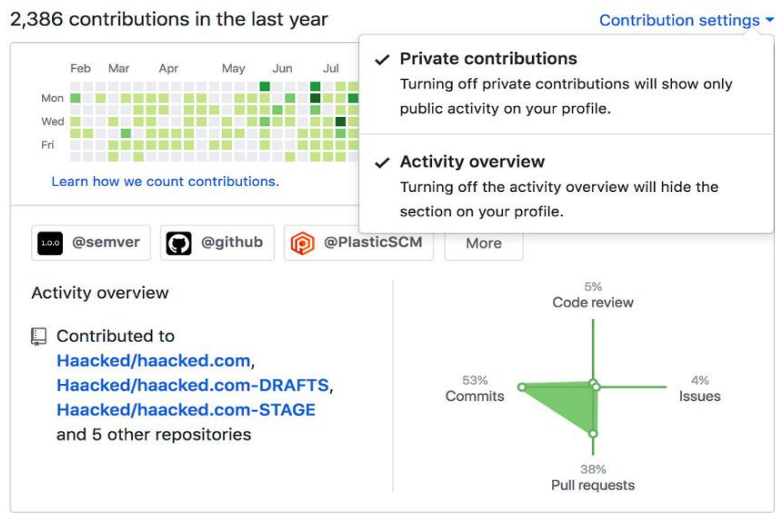


图 16-4 展示私有贡献与活动图

### 16.1.6 贡献活动

贡献图下方是贡献活动时间线。用户在 GitHub 上的活动时间线可以追溯到第一次提交。

注意：某些情况下，有的用户在加入 GitHub 之前，活动时间线中就有活动记录。甚至有的用户可能在 GitHub 诞生之前就有活动记录！这是因为活动时间线包括基于 Git 提交时间戳的仓库中的 Git 活动。在 GitHub 诞生之前所创建的仓库也会被导入到 GitHub 中。

## 16.2 Star 仓库

提示：当访问一个有价值的仓库时，用户可以点亮仓库页面右上角的星星。Star 不仅是对仓库所有者点赞，还可以充当仓库书签。

在个人页面上，单击“Stars”选项卡以查看 Star 仓库的列表。任何访问个人页面的用户都能查看该列表。探索其他用户 Star 的仓库可以发现许多有趣的新项目。

提示：<https://profile-summary-for-github.com/user/haacked> 个人页面利用可视化方式展示了该用户哪些仓库拥有最多的 Star 量（见图 16-5）。其他用户只需将链接中的用户名替换为自己的用户名即可查看自己的个人页面。



图 16-5 可视化个人页面

## 16.3 关注用户

当用户访问其他用户的 GitHub 个人页面时，个人图片下方会出现一个“Follow”按钮。单击“Follow”按钮可以在仪表板中订阅有关用户活动的通知。所关注的人也会进入 GitHub 的推荐系统。例如，如果用户关注的某个人 Star 了某个公共仓库，则该仓库可能会被推荐并显示在仪表板的“Discover repositories”中。

可以通过单击主页的“Following”选项卡来查看该用户所关注的人，还可以通过单击“Followers”选项卡来查看所有关注自己的人。

## 第 17 章 参加活动

本章重点学习：

- (1) 通过活动促进职业发展
- (2) 查找所在地区的活动
- (3) 在活动中发言

作为刚入行的软件开发人员，工作初期是很有挑战性的，特别是一个人单打独斗时。这就是为什么 **GitHub** 对软件开发人员来说是一个如此重要的工具。**GitHub** 不仅是世界上最大的代码托管平台，也是最大的软件开发社区。参与 **GitHub** 可以和更多人建立联系。

大多数使用 **GitHub** 的人只是对 **GitHub** 提供的服务略知一二。通过阅读本书，你将比许多其他开发人员领先一步。了解如何使用 **GitHub** 的确很有帮助，但是在 **GitHub** 上还有很多探索的事情。

在本章中，我们鼓励读者暂时远离键盘并亲自与其他开发人员会面，从而让职业生涯再领先一步。参加活动是建立人际关系的好方法，对整个职业生涯都有好处。在活动中发言是职业生涯中进一步发展的方法。这可能会让人望而生畏，但每个人的发言都是有价值的，即使是那些新手。

### 17.1 探索不同类型的活动

许多类型的活动都集中在软件开发人员身上。它们的范围从非正式聚会或用户组到有组织的多日国际软件会议。在本节中，我们将介绍最常见的活动类型以及每种活动类型的预期内容。

#### 17.1.1 小型聚会和用户组

小型聚会（**meetup**）和用户组（**user group**）是开发人员围绕某一主题进行的非正式聚会。许多聚会每月会安排一次，由当地公司或软件兴趣小组主持。这些活动是初次尝试的好去处。它们往往是某一地区的小型聚会。每个月都有一位当地演讲者谈论与该小组相关的话题。（一些活动有时会从外部邀请演讲者，但通常他们会重点关注当地的演讲者。）

建议：**Meetups.com** 是一个查找聚会的好方法。如需要查找华盛顿西雅图的 **JavaScript** 聚会列表，则可以访问以下网址：

<https://www.meetup.com/topics/javascript/us/wa/seattle>

或者可以在 **meetups.com** 上按地区搜索聚会。

本地聚会的一些例子包括：

- (1) **Brooklyn JS**：位于纽约布鲁克林，见 <http://brooklynjs.com>；



(2) .NET São Paulo: 位于巴西圣保罗, 见 [www.meetup.com/dotnet-Sao-Paulo](http://www.meetup.com/dotnet-Sao-Paulo);

(3) SD Ruby: 位于加利福尼亚州圣地亚哥, 见 [www.sdruby.org](http://www.sdruby.org)。

### 17.1.2 区域会议

区域会议 (regional conference) 是一个相对较小的会议, 虽然也欢迎本地区以外的演讲者和参会者, 但会议的重点是为本地开发人员和演讲者提供一个联系和展示他们工作的场所。

这些会议通常是一天或两天。这些会议一般只有一个 track, 最多两个。它们在规模和组织上比聚会更上一层楼, 通常每年举行一次, 而不是每月举行一次。他们中的一些人经常在会议之前或之后提供研讨会。这些研讨会通常需要额外付费, 但会针对特定技能或技术提供更深入的培训, 例如一个专门用于提高 Git 技能的全天研讨会。在能负担得起并且内容对你有所帮助的情况下, 研讨会是个很值得的投资。一些当地会议的好例子包括:

(1) .NET Fringe: 位于俄勒冈州波特兰市, 见 <http://dotnetfringe.org>;

(2) JSConf Hawaii: 位于夏威夷檀香山, 见 [www.jsconfhi.com](http://www.jsconfhi.com);

(3) GoRuCo: 位于纽约, 见 <http://goruco.com>。

### 17.1.3 黑客马拉松

黑客马拉松 (hackathon) 与会议不同。会议更侧重于让演讲者通过演讲来教授一个主题, 而黑客马拉松则侧重于构建。黑客马拉松是一项可能持续数天的活动, 其中一群人组成团队共同编写代码以解决某种问题。黑客马拉松通常的形式是先提出某种问题, 然后团队的任务是构建解决方案。他们可能使用的技术栈通常是取决于这次黑客马拉松关注的技术。例如, 移动开发黑客马拉松将要求参与者构建一个移动应用程序来解决问题。

Hackathon 是 hack 和 marathon 两个词的合成词。许多人通过让团队夜以继日地工作以及减少睡眠, 从而将马拉松方面发挥到极致。还有一些人则试图通过强迫参赛者离开工作场所以此来平衡工作时间和睡眠时间。

提示: 黑客马拉松通常非常包容初学者。例如, [www.sdhacks.io](http://www.sdhacks.io) 对世界上任何 18 岁或以上的高中生或大学生开放。报名参加黑客马拉松时并不需要一定有一个团队, 有时也可以现场组队。关于这一点, 最好查看特定黑客马拉松的常见问题解答以了解更多细节。

全球最大的黑客马拉松之一是针对大学生的。这是微软创新杯 (见 <https://imaginecup.microsoft.com/en-us/Events?id=0>)。创新杯的获胜者可以获得萨蒂亚·纳德拉 (微软首席执行官) 的指导, 前往参加世界锦标赛, 以及获得 Azure 赠送的礼物和 100,000 美元。

参加黑客马拉松是了解新技术的好方法。黑客马拉松的目标不是设计和实现最终产品，而是将想法一点一滴的拼凑在一起。最终产品应该看起来更像是一个原型，而不是一个经过润色的应用程序。很多时候，黑客马拉松会有很多了解特定技术的导师来指导。我们应该将黑客马拉松视为一个进行实验和学习的活动。

提示：虽然参加正式的黑客马拉松会提供导师、空间等资源，有时还会有奖励，但我们也可以随时与朋友聚在一起，自己举办一次黑客马拉松！只需要选择一个时间、地点和目标，然后尝试将想法拼凑成一个原型！试一试也无妨！

#### 17.1.4 大型会议

大型会议往往规模很大，会吸引来自全国各地（世界各地）的参会者。出席人数往往以千计。参加这些会议需要更多的前期计划。这不仅仅是安排您的航班和酒店。这些会议往往有许多议题，所以对于任何给定的时间段，您可能必须从五个或更多同时举行的演讲中选择要观看的演讲。

像区域会议一样，大型会议通常会在会议之前或之后提供一系列研讨会。除了研讨会之外，许多会议期间还包括实践环节（hands on labs）。实践环节通常包含在会议的费用中，并提供了一个很好的机会来实际尝试在会议上见识到的技术。许多此类会议都是由大型科技公司举办的，比如微软的 Build 会议和苹果的 WWDC。一些例子如下：

- （1）Oscon：位于俄勒冈州波特兰，其链接为  
<https://conferences.oreilly.com/oscon/oscon-or>;
- （2）Build：会议举办地点每年都在变化，其链接为  
[www.microsoft.com/en-us/build](http://www.microsoft.com/en-us/build);
- （3）WWDC：位于加利福尼亚州旧金山，其链接为  
<https://developer.apple.com/wwdc>。

### 17.2 了解在活动中会发生什么

不同类型的活动在内容上可能存在巨大差异，但也有很多共同点。最大的共同点就是其他开发者也会到场。对于不能常与开发者交流的人来说，经常会感到孤独，因此对于那些真正关心软件技术并提升自己的人来说，参加活动是一个结识志同道合的人的好机会。

#### 17.2.1 主题演讲

许多会议，尤其是大型会议，都会包括主题演讲。有些会议会包括很多主题演讲。主题演讲为会议定下基调，通常与会议的主题相关。对于多主题会议，主题演讲期间通常不会安排其他演讲。

提示：对于由微软或苹果等大型软件公司举办的大型会议，他们通常会在主题演讲会上发布新产品和新功能。

### 17.2.2 会议 track

会议的主要吸引力在于会议的 track。一个 track 是一组谈话，通常围绕一个主题组织。较小的会议可能只有一个 track，而较大的会议可能有大量 track。小型聚会（meetup）和用户组（user group）可能只有一个谈话。

根据会议类型的不同，每次会议时长为 30 分钟到 75 分钟。其中许多会议结束时会有一些时间供观众提问和回答（Q&A）。

警告：如果您参加问答环节，只利用提问时间做陈述是不礼貌的行为。确保你的问题以问号结尾。

有时，演讲者会要求在演讲结束后再与参与者进行更深入的交流。这并不是因为他们不想在公共场合上回答提问。通常情况下，问答环节只持续 5 到 15 分钟，演讲者总是没有足够的时间来完全理解问题并提出一个有效的答案。参与者可以在演讲结束后直接与演讲者会面（不过至少应该让他们把笔记本电脑和其他东西从台上拿下来），或询问他们在会议期间何时可以一起喝杯咖啡（如果这是一个多日会议）。也可以询问是否有他们更喜欢的异步跟进方式。

牢记：会议主持人通常也会参加会议以学习一些东西；他们不仅在那里进行主持，所以也要尊重他们的时间。

### 17.2.3 Hallway tracks

Hallway track 也是很有价值的一个环节。通常在会议日程表上找不到 Hallway track。它是人们创造的一个术语，指的是在会议间隙、午餐期间在走廊上进行的非正式对话。有时，参会者会遇到某个话题的世界级专家与几位与会者进行随意交谈，他们很欢迎其他人加入他们的对话。这些 Hallway track 通常比参加演讲更具教育意义，因为参会者可以参与对话，而不仅仅是倾听。它们也可能会带来终生的友谊和有趣的合作。

提示：在会议上结识新朋友可能会让人感到胆怯，尤其对于内向的人。其实大多数人都会有这种感觉。如果你鼓起勇气向一个不认识的人介绍自己，她可能会感到宽慰。如果你是一个外向的人，寻找机会用一种低压力的方式吸引内向的人来加入谈话。

### 17.2.4 会后的会议活动

许多会议包括会后的活动。根据会议的规模，其中一些活动可能会相当奢侈。

警告：通常，这些活动包括喧嚣的音乐和酒精，所以如果你不喜欢这些东西，自己要谨慎。

许多会议都试图让与会者的活动更具创造性和包容性。例如，一个会议可能会租借一个有很多乒乓桌的乒乓球场地。

### 17.2.5 受人尊敬的专业环境

在大多数活动中，您可以期待一个尊重、包容和专业的环境。这种环境有利于人际交往、社交和学习。

在某些情况下，会议未能达到这种预期。因此，许多会议现在采用并执行行为准则。行为准则概述了会议参与者的行为期望。更重要的是，它向那些经常成为骚扰目标的人传达了这样的信息：大会非常重视骚扰问题，骚扰是不受欢迎的。

## 17.3 熟悉 GitHub 活动

考虑到本书是关于 GitHub 的，我们前面忽略了一些关于 GitHub 自己的活动信息。其实 GitHub 全年举办和赞助多个会议。要了解哪些活动即将举行，可以访问 <https://github.com/events>。此页面列出了 GitHub 自己即将举行的活动以及它赞助的活动。

### 17.3.1 GitHub Universe

GitHub Universe（见 <https://githubuniverse.com/>）是 GitHub 的官方会议。该活动每年在 GitHub 总部所在地加利福尼亚州旧金山举行。会议通常在 10 月或 11 月左右举行。

正如 GitHub 所描述的那样：

GitHub Universe is a conference for the builders, planners, and leaders defining the future of software（GitHub Universe 是一个为定义软件未来的构建者、规划者和领导者举办的会议。）

GitHub 通常会在该会议主题演讲期间发布年度事宜。它吸引了来自诸多著名软件公司的知名演讲者们。

2018 年，大会约有 1800 名与会者，大会包括三个 track。会议费用大约 99 美元，如果您还要参加研讨会，则为 199 美元。

### 17.3.2 GitHub Satellite

GitHub Satellite 会议（见 <https://githubsatellite.com/>）是 GitHub Universe 的一个分支。他们将 GitHub Universe 风格的会议带到了世界各地。

GitHub Satellite 会议每年举办一次，往届的 Satellite 会议曾在柏林、东京和伦敦等地举行。

### 17.3.3 GitHub Constellation

GitHub Constellation（见 <https://githubconstellation.com/>）是一系列小型社区活动，每年在全球举办多次。这些活动侧重于当地社区，并经常邀请当地的演讲者参加。它们通常是免费的，举办时间一般是一个或两个晚上，而不是像 Satellite 和 Universe 那样的全天会议。

### 17.3.4 Git Merge

Git Merge（见 <https://git-merge.com/>），是一个由 GitHub 赞助的会议，但关注的是 Git 版本控制工具和 Git 软件用户。GitHub 这样介绍它：

Through technical sessions and hands-on workshops, developers and teams of all experience levels will find new ways to use, build on, and scale Git.（通过技术会议和实践研讨会，所有经验水平的开发人员和团队都将找到使用、构建和扩展 Git 的新方法）。

该会议在正式会议前特别举办了一场实践研讨会，重点讨论一系列 Git 相关的主题。这是了解有关 Git 的更多信息并提高 Git 技能的绝佳机会。

## 17.4 在活动中发言

许多开发者都有这样的误解，即只有从业多年的专家或者性格外向的人才可以在会议上发言，但事实并非如此。

### 17.4.1 每个人都有故事要讲

无论是否意识到这一点，在每个人的内心深处，都有一个有趣的故事要讲，即便是软件开发新手。所有此类活动的共同点是，它们鼓励持有不同观点的人的思想碰撞。例如，有经验的开发人员在长时间使用一项技术时，通常会失去作为初学者的思考。一个初学者关于他们如何努力学习某一特定技术的演讲通常会让听众们大开眼界，而且对有经验的开发人员来说，初学者的经验教训会督促他们做得更好。

### 17.4.2 成为演讲者的好处

成为演讲者有很多好处。主要是演讲者会花时间去深入学习一个主题。如果你打算发表一个主题演讲，最好的方法是在已经有所了解的基础上进行研究。把一个主题教给别人是巩固自己理解的好方法。不仅如此，这还是一个很好的获取反馈意见的途径。通常在演讲后，听众会分享他们独特的见解。如果不分享，就不会收到这些反馈。还有一个好处是，作为演讲者，可以获得足够的曝光和交流机会。因为人们更想和“演讲者”交流。当你在一些会议上发言时，你的名字就会出现在那里。当你申请一份新工作时，人们可能会给你更多的机会，因为他们听过你的演讲。

演讲者还有一些额外的好处。通常，会议有仅限演讲者参与的活动，演讲者们可以相互了解。这种活动通常会带来很好的社交机会和友谊，因为在较小的环境中更容易记住人。此外，随着在越来越多的会议上发言，你可以认识很多发言者，结实很多新朋友。当参加一个新的会议时，如果遇见曾经一起演讲的熟人，那么会很有帮助。



## 17.5 为活动寻找资金

一些活动和会议比较昂贵。Microsoft BUILD 的费用可能高达 2,500 美元，Google I/O 的费用也接近 1,200 美元，另外还需要前往会议的交通费和停车费以及可能产生的酒店和航班费用。

如果会议是一个宝贵的学习机会，许多软件公司会花钱让员工参加会议。如果你在软件公司工作，可以咨询一下老板是否可以报销费用。为了让理由更充分，可以解释一下在会议上学到的东西将如何提高工作效率。

大型软件公司可能也会赞助活动，需要志愿者参加并帮助运营。有些公司甚至会给参加会议的员工提供津贴，作为员工薪酬的一部分。

另一种资助参会的方法是申请奖学金。Grace Hopper Celebration 是一个大型会议，有超过 1.8 万名与会者，旨在鼓励计算机领域的女性。这个会议有技术专题，也有关关注多样性和包容性的专题，会议通常持续三天，地点在美国各地。举办会议的 Anita Borg 研究所还提供奖学金，您可以在 <https://ghc.anitab.org/2019-student-academic/scholarships> 上找到。这些奖学金通常包括机票、住宿费、交通费、餐费和参加活动的门票。他们专注于让学生和教师获得这项特殊奖学金。

大多数会议将免除演讲者的会议门票费用。许多会议还将包括酒店和交通费用。有些会议甚至会提供除费用外的酬金。这些完全取决于会议。此外，有些会议会邀请志愿者来帮助组织会议，这往往还会带来一些额外的好处，比如免费门票，这样当志愿活动空闲时，就可以享受会议的其余时间。Grace Hopper Celebration 将其称为 Hopper Program，您可以在 <https://ghc.anitab.org/hoppers> 上找到该计划。

牢记：如果你发现了一个特别想参加的活动，在自己掏出几千美元之前，可以先咨询周围的人！询问你的人际关系网中是否有人知道是否可以申请奖学金，申请成为演讲家或志愿者，并询问你的公司或学校有哪些资源可供使用！