## bot机制分析

Bot是一种在互联网上执行自动化任务的软件应用程序，开源社区Bot基于Webhook机制和平台API实现，基于用户个性化需求，扩充平台的功能以便于项目的开发、维护和管理。开源社区Bot可自动回应用户，执行Issue、评论等相关的系列操作，利用其自动化工作流程，可以使开发者摆脱冗余、繁杂的工作，提高开发效率，进而促进社区繁荣。

### Bot开发的基石——Webhook机制

在开源社区中，bot的工作方式通常是这样的：项目中某一事件状态发生改变，bot检测到这一改变，然后获取相应的事件数据进行处理，最后自动执行相关操作。Webhook就是一种服务于这个过程的基础机制。

Webhook，类似于异步编程中的"订阅-发布模型"，是一种通过回调函数，去增加或者改变网页功能和行为的方法。这些回调函数通常由与网站原始开发没有关联，而是由第三方用户和开发者来定义、修改和管理。

Webhook允许用户在项目设置中订阅一些事件，当有项目中订阅的事件触发后，平台会向Webhook配置中的URL发送一个HTTP荷载，其中包含该事件的相关数据。这个URL就是bot运行的服务器地址，bot接收到相关事件数据后就能进行一系列自动化操作。

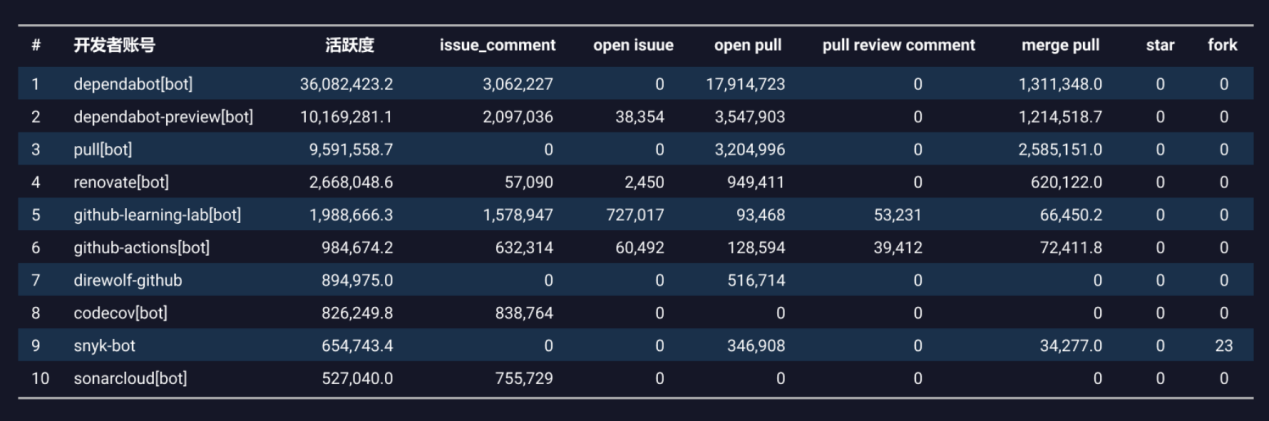
事件，在进行Webhook配置时，用户可以根据个人需求订阅自己想要监听的事件。例如，你想要在合并请求事件发生后，提醒相关负责人处理，可以订阅合并请求事件。Webhook默认情况订阅push事件，用户可自行选择订阅全部事件或者自定义部分事件。

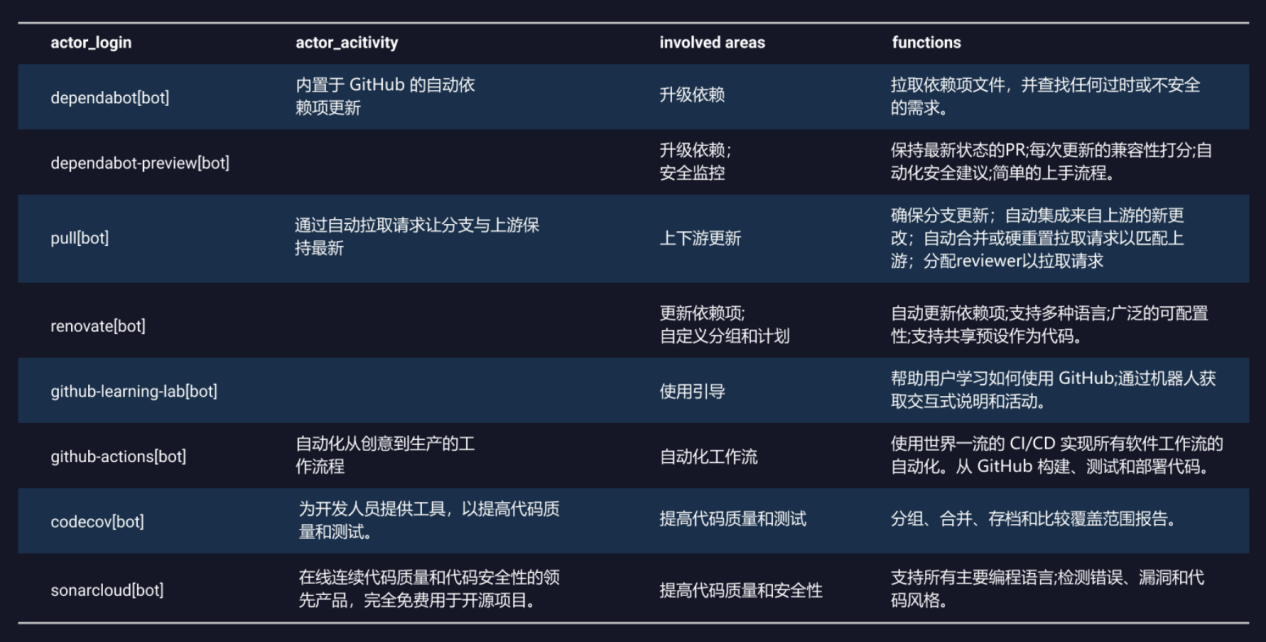
### Bot的应用程序——GitHub Apps

2016年，GitHub官方推出了一类产品Apps，它自身便拥有GitHub提供的认证信息，可通过调用一系列GitHub API来完成相关操作，而无需借用其他的用户身份来运行。事实上，GitHub Apps本身就是一类GitHub用户，用户名以‘bot’结尾与普通用户加以区别，这有助于我们在大量日志中将其筛选出来。

GitHub Apps可以直接安装在组织和用户帐户上，获得对特定仓库的访问权限，它拥有内置的 webhook和精细、具体的控制权限。事实上，由于在项目管理过程中很多自动化的需求是相同或者类似的，因此很多GitHub Apps的开发时重复冗余的。为了减少冗余开发、提高效率，GitHub提供了一个Apps市场，里面包含了广大开发者开发的各种类Apps。开发者可以很方便地在找到自己需要的Apps，也可以分享或者出售自己满意的Apps。

根据2020年GitHub数字洞察报告，全域活跃排名最高的10个开发中中有8个账号为GitHub Apps，其余两个也是自动化协作相关的开发者账号，可见Bot应用和GitHub Apps的流行程度。





### Bot的开发框架——以Probot为例

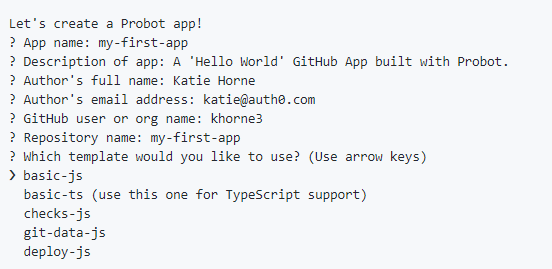
在进行GitHub Apps的开发时，需要进行各项基本信息的填写与大量关于仓库、用户权限的设置，这些设置对于初次接触bot的开发者是不够友好的，并且大量琐碎的配置工作耗费了开发者较多的事件和精力。因此，一个易于Bot构建的开发框架是非常有必要的。

Probot是一个基于Node.js环境构建 GitHub Apps的框架。通过Probot框架开发GitHub Apps，用户可以不理会那些琐碎的webhook接收、验证等工作，专注于对应用功能的开发。

基于Probot构建的应用程序易于编写、部署和共享，目前GitHub平台上已积累了许多受欢迎的Probot应用程序。

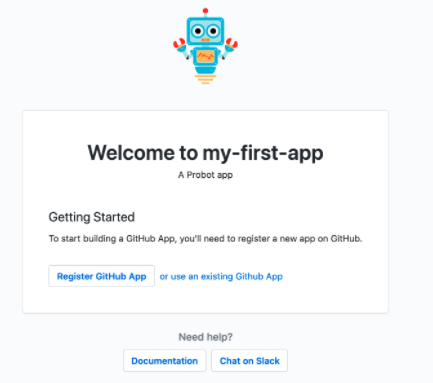
通过Probot构建GitHub Apps的命令和信息配置界面如下：





通过Probot构建GitHub Apps生成的文件目录和Web页面如下：





### 自动化工作流的另一机制——GitHub Actions

GitHub 2018年推出的Actions，可帮助用户自动完成软件开发周期内的任务。在仓库的GitHub Actions中可自动化、自定义和执行软件开发工作流程。 此外，还可以发现、创建和共享操作以执行您喜欢的任何作业（包括 CI/CD），并将操作合并到完全自定义的工作流程中。

GitHub Actions基本概念：

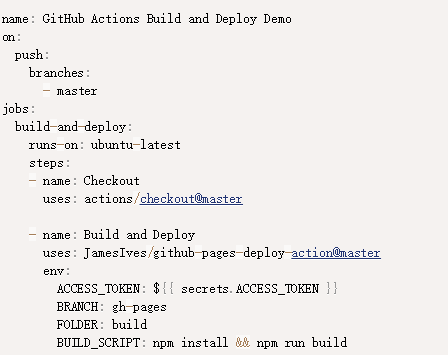
1. workflow ：持续集成一次运行的过程，就是一个workflow；

2. job：一个workflow 由一个或多个 jobs 构成，含义是一次持续集成的运行，可以完成多个任务；

3. step：每个 job 由多个 step 构成，一步步完成；

4. action：每个 step 可以依次执行一个或多个命令(action)。

一个简单的GitHub Action构建与部署demo如下：



在master分支发生push事件时触发；

只有一个job，运行在虚拟机环境ubuntu-latest：

第一步是获取源码，使用的action是actions/checkout；

第二步是构建和部署，使用的action是JamesIves/github-pages-deploy-action。

需要四个环境变量，分别为GitHub 密钥、发布分支、构建成果所在目录、构建脚本。其中，只有GitHub密钥是秘密变量，需要写在双括号里面，其他三个都可以直接写在文件里。