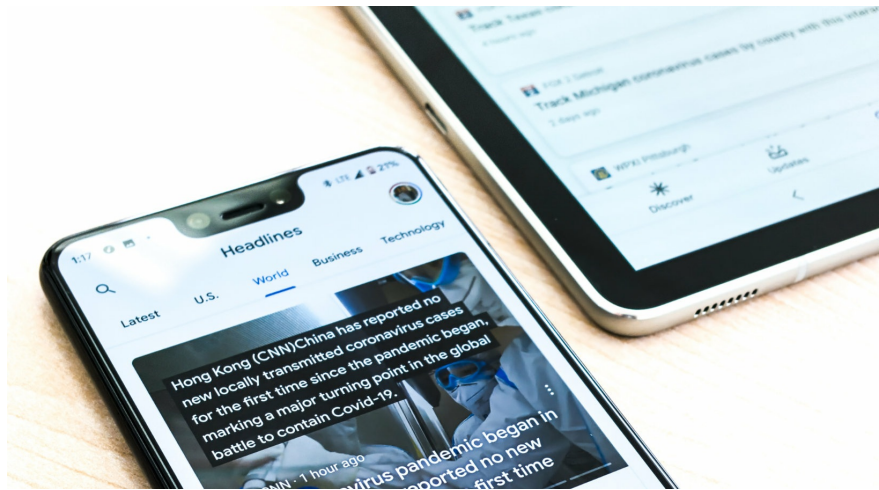


使用Railway和Miniflux零成本搭建RSS服务

Via Cysime Moflu



RSS（聚合内容），一种“古老”的消息来源格式，古老到连著名的RSS聚合器兼阅读器Google Reader都快停止服务十年了。但近几年随着开源项目RSSHub发布极大程度上丰富了可订阅内容，加第二春。我也是从那时候开始重新依赖RSS获取信息。

想要使用RSS作为自己的信息获取工具，就一定需要一个RSS聚合工具，将你所感兴趣的信息源全部集中到一处，通常可以使用下面的几种方案：

- 本地应用程序（如Reeder），但这种方案无法原生多设备同步，订阅源和阅读状况只在单台设备内存储，必须配合商业网络服务或自建服务来跨设备同步内容。
- 商业订阅服务（如Feedly、Feedbin），但大多数需要每月支付一定的订阅费用。
- 自建服务（如FreshRSS、tt-RSS、Miniflux），需要花一定的心思自行搭建，大多需要自行准备服务器，同样需要一定的物质（金钱）成本。

RSS聚合软件对比

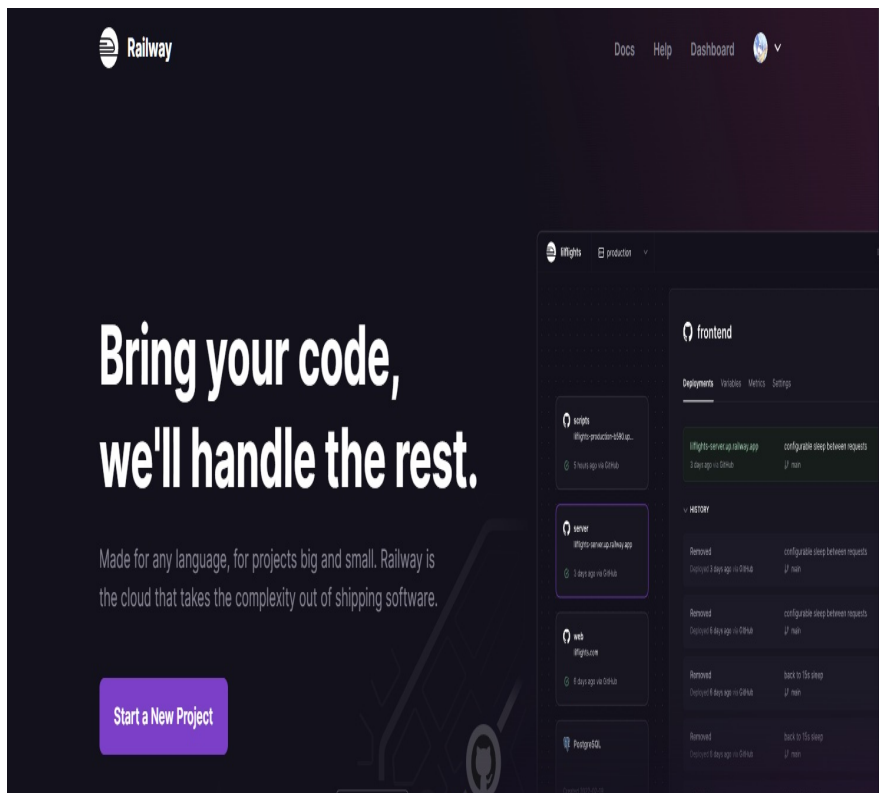
今天我们要讨论的是第三种方案，目前常用的自建RSS聚合器框架主要有：

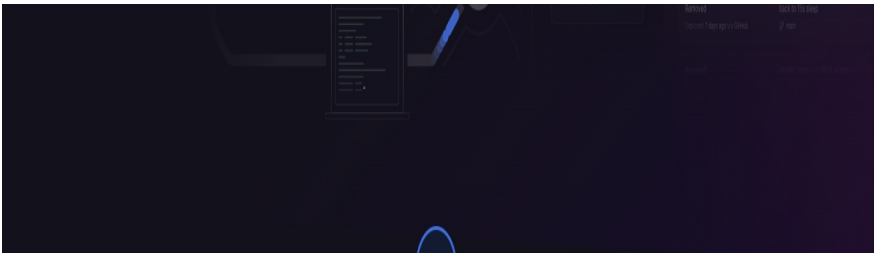
- [FreshRSS](#)，功能强大、易于使用，基于PHP和SQL数据库
- [Tiny Tiny RSS](#)（tt-RSS），功能非常强大、扩展性极强，基于PHP和SQL数据库
- [Miniflux](#)，极度简约但应有的功能一个不缺，基于Go语言

可以看出，前两个框架虽然功能上更强大且搭建起来比较简单（可以使用Docker，或者直接使用PHP虚拟主机搭建）但受限与PHP+SQL的组合，整个框架都显得比较“重”或者慢，且无法做到真正事实上Miniflux官方就给出了[使用Heroku搭建的教程](#)，但Heroku免费版会定时休眠且经过测试发现无法使用ping工具保持真唤醒状态（为什么说“真唤醒”？因为此时在Heroku控制面板中该服务并没

Railway：不休眠的PaaS平台

虽然本文的目的是0成本搭建，但也并不意味着我们就要因此妥协使用体验。这时候我们可以把目光转向另一个PaaS平台：[Railway](#)（该链接含Referral Code）





这是一个2020年成立的PaaS平台，无需信用卡的免费版可以提供5美元的额度，如果绑定信用卡则每个月可以使用10美元的额度，用户仅需支付超过该额度的使用费，对于普通个人用户而言，5美元——更重要的是，它支持数据库和Docker，所以比起Vercel或Netlify，它更像是Heroku的替代品，更优质的替代品。这就为我们达成0成本搭建RSS聚合器且不损失使用体验带来了可能。

动手搭建

说了这么多，现在让我们开始动手搭建基于Miniflux+Railway的专属于自己的RSS聚合器。

需要注意的是，为了让更多人能够自行搭建服务，因此本文中所有的操作都尽量使用有图形化的程序并使操作尽可能简单。

准备工作

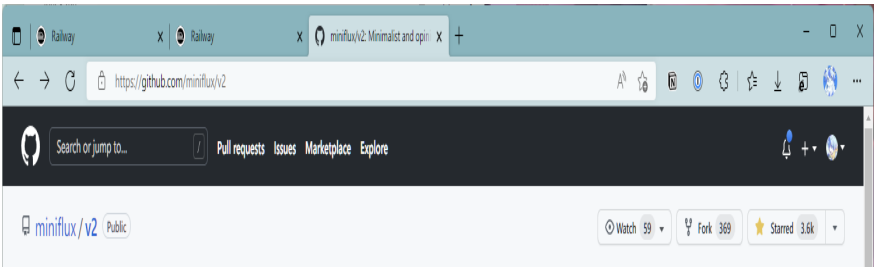
正式开始之前，我们需要准备：

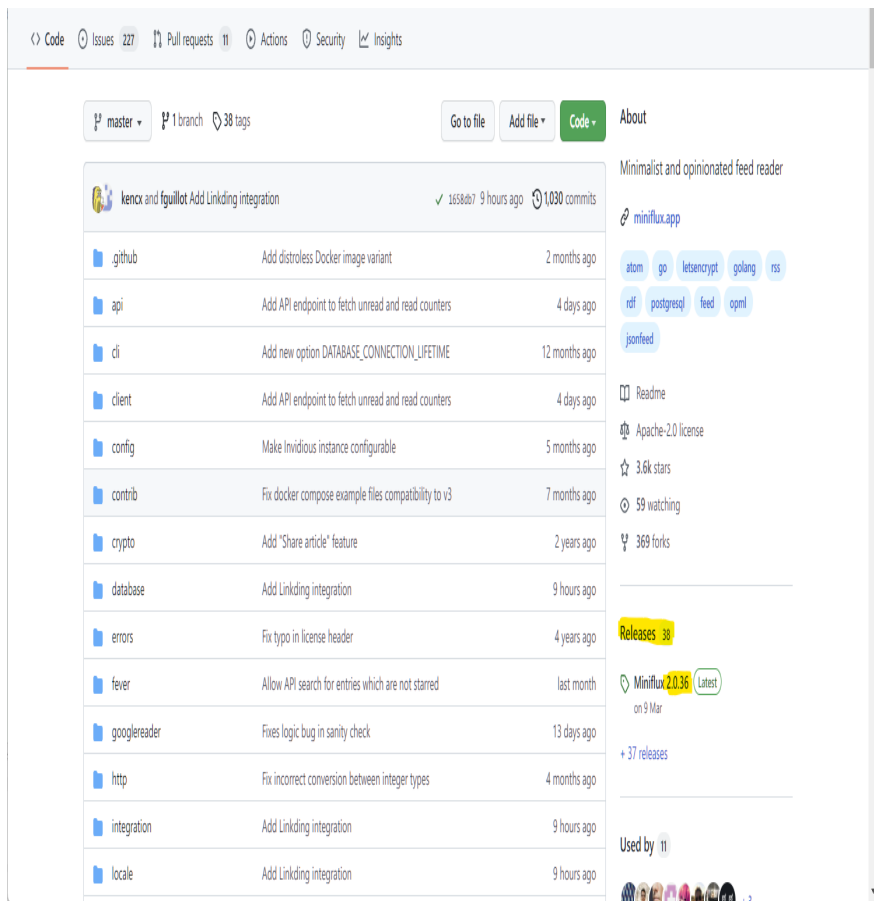
- [GitHub](#)账号
- [Railway](#)账号
- [GitHub Desktop](#)客户端（可选，图形化管理更加简单，或者你可以直接使用Git）
- 一台电脑

首先我们需要去[Miniflux v2](#)的[GitHub 页面](#)并Fork该项目到自己的账户中。完成后打开并登录GitHub Desktop客户端， 点击顶部菜单栏的“File - Clone a repository”，找到自己刚刚Fork的项目，选择

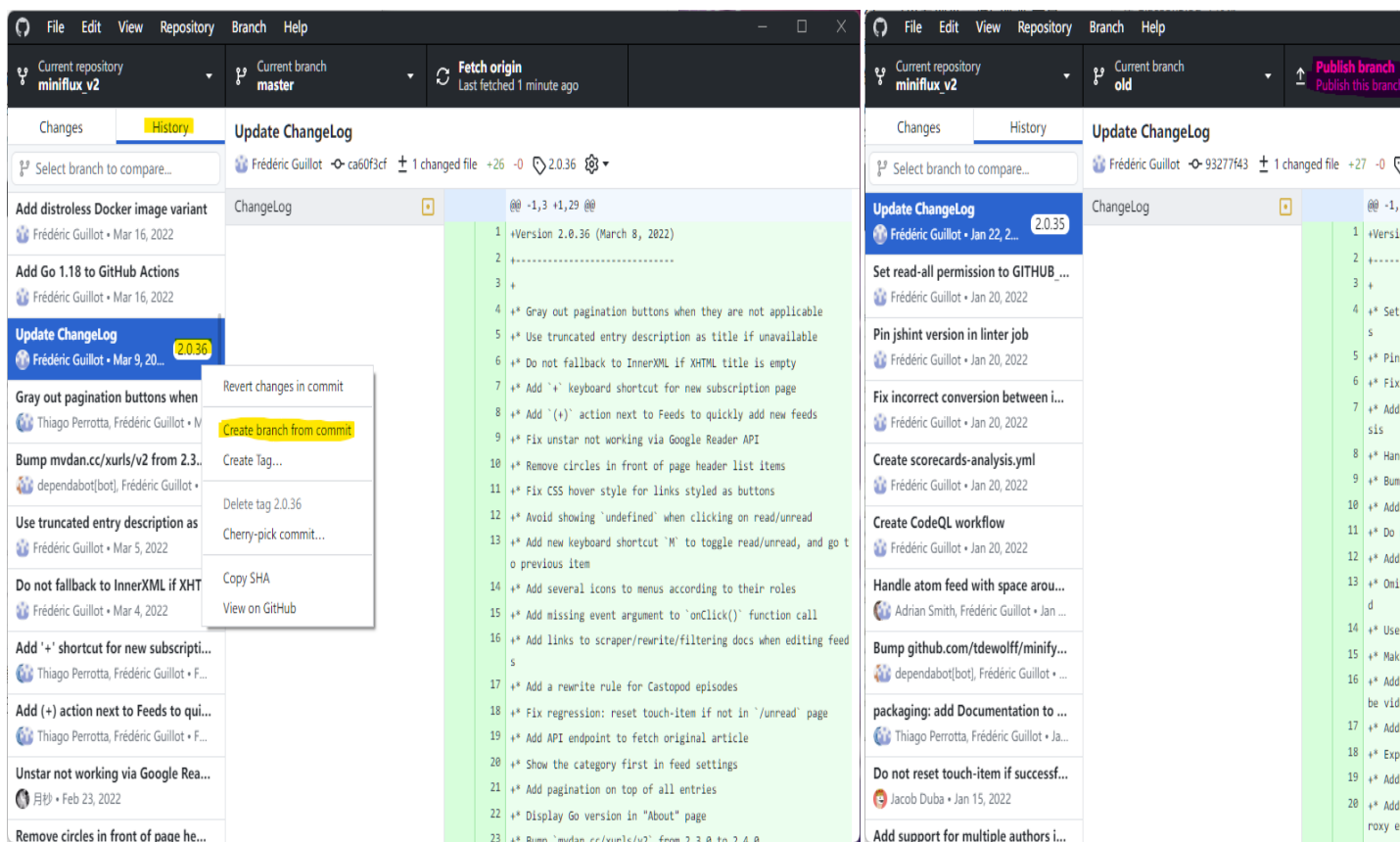
创建稳定版分支

由于Miniflux的默认分支（master）本身其实是不稳定的开发版，直接部署使用开发版多多少少会遇到各种问题，为了保证服务的稳定性和可用性，故我们还需要单独创建一个稳定版分支。



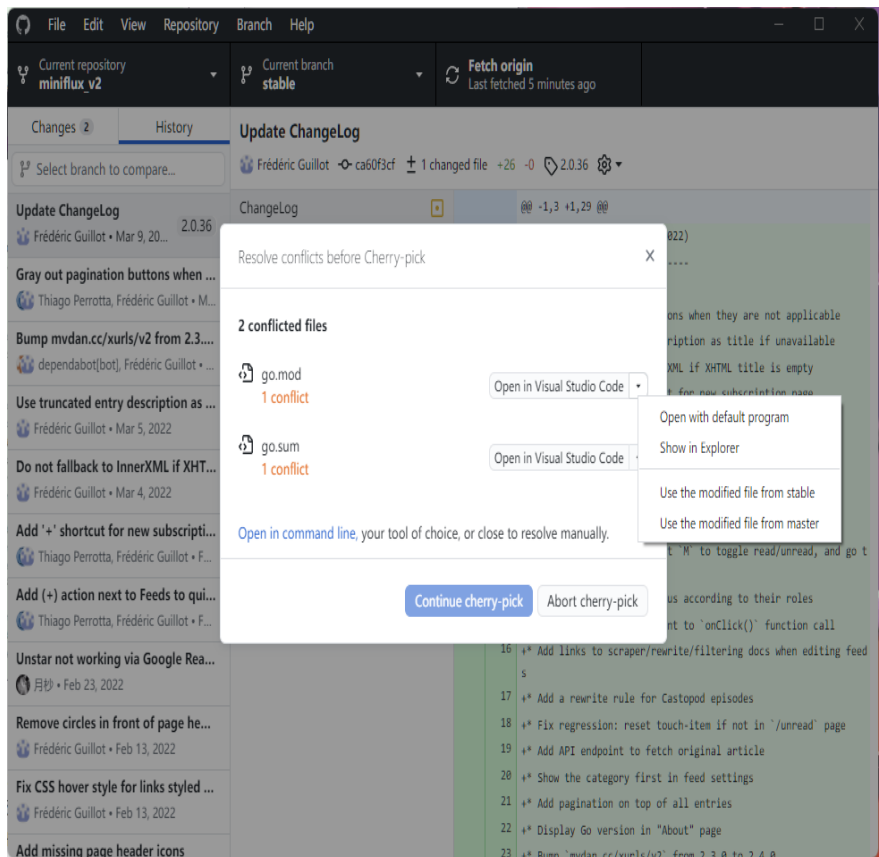


让我们回到Miniflux的GitHub页面，注意上图标黄的“Release”部分，可以看到当前稳定版版本号为2.0.36，这也是该Release对应的tag，记住这个版本号，在GitHub Desktop打开下载到本地的项目分支stable并Create branch，完成后，顶部的第三个大按钮会变成“Publish branch”，点击它使之同步到GitHub仓库中。至此，稳定版分支创建完成。



更新稳定版分支

如果要更新你的稳定版分支，做法也很简单，首先确定Miniflux已经发布了新的稳定版，打开GitHub Desktop，点击客户端窗口上的“Current branch”，选择分支master，这时右边的大按钮会变成“Publish branch”，点击它使之同步到GitHub仓库中。至此，稳定版分支创建完成。

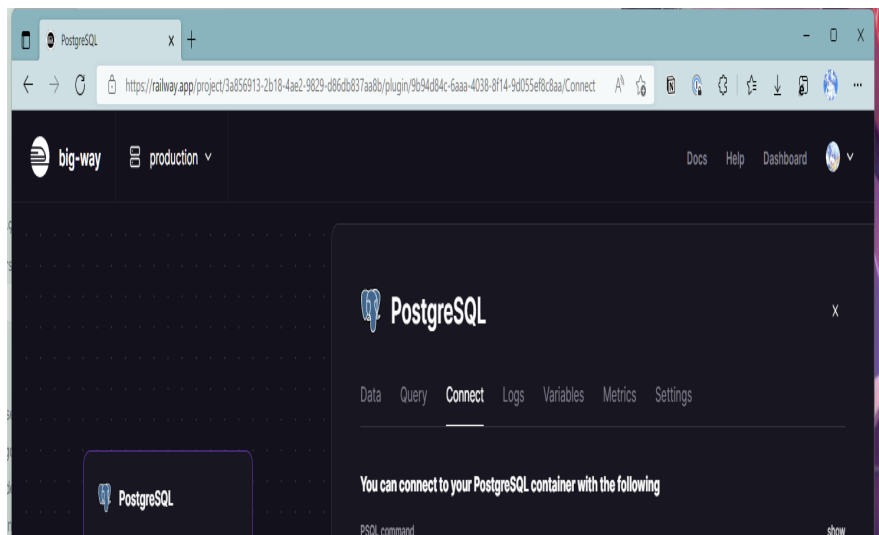


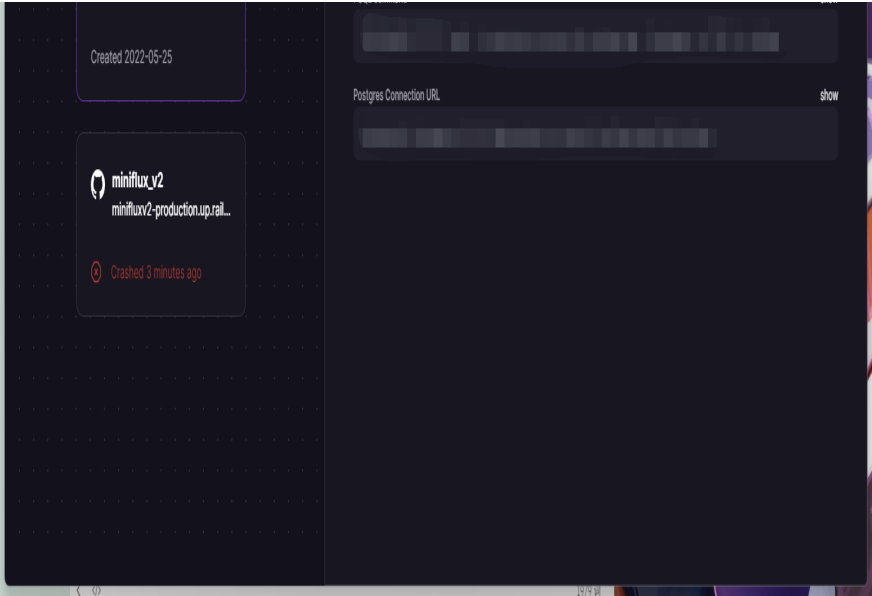
如果出现冲突（conflict），则会弹出上面的窗口，由于我们根本就不会动任何源代码，故直接用新版覆盖即可：点击右侧的三角选择“Use the modified file form master”并Continue cherry-pick即可里面自动部署新版。

部署到Railway

现在我们可以将稳定版分支部署到PaaS平台，比如我们接下来会用到的Railway。

进入Railway控制面板，新建项目（New Project）- Deploy from GitHub repo - 绑定GitHub账号并选择我们在上一节做好稳定版分支的项目，这时系统会自动开始部署，因为我们还没有设置数据库PostgreSQL，稍等片刻后点击该数据库，导航到“Connect”选项卡，找到“Postgres Connection URL”，复制备用。





接着，回到刚刚连接的GitHub Repo模块，点击它，在右侧弹出的窗口，导航到“Settings”选项卡，将“Deployment Trigger”下面的选项从master改为stable，即切换为我们在上一节里面增加的域名。

完成后切换到“Variables”选项卡，点击右侧的“RAW Editor”，将下面的变量按需修改并去掉注释后复制粘贴到文本框内，点击Update即可。

```
1 2 3 4 5 6 7 DATABASE_URL=刚刚复制留存的PostgreSQL链接 PROXY_IMAGES=all RUN_MIGRATIONS=1 BASE_URL=https://域名,如果你无法自备域名则可以直
```

提交了环境变量后Railway会自动重新部署，可以在“Deployments”选项卡里面查看进度。完成后即可登录使用。Miniflux原生支持Fever和Google Reader API，可以在Miniflux的设置中启用，这样则至此，一个简约但不简单的0成本RSS聚合器就搭建完成了！快去订阅自己喜欢的源享受干净可控的阅读环境吧！

扩展阅读：RSSHub

有了自己的RSS聚合器后，剩余的工作就是寻找并订阅自己喜欢的内容（源）了，这里推荐使用RSSHub，让各种原本无法使用RSS的平台支持RSS，可以极大丰富订阅内容。RSSHub官方可以阅读这篇文章了解如何使用Vercel免费搭建一个自己的RSSHub。