# github星标超3万！Postman最强平替Bruno你用了吗？

## 序言

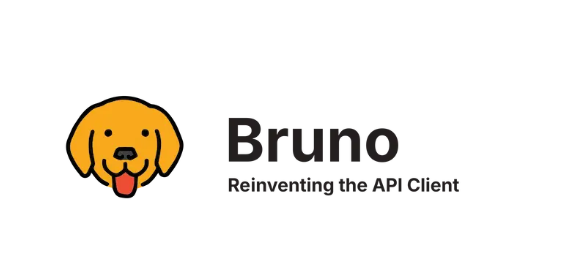
Postman长期以来都是API测试领域居领先地位的测试工具。以其友好的工具界面，便捷的API接口请求编辑，强大的脚本支持和云端服务，以及丰富的生态，在接口开发和测试中，有极为广泛你的应用。



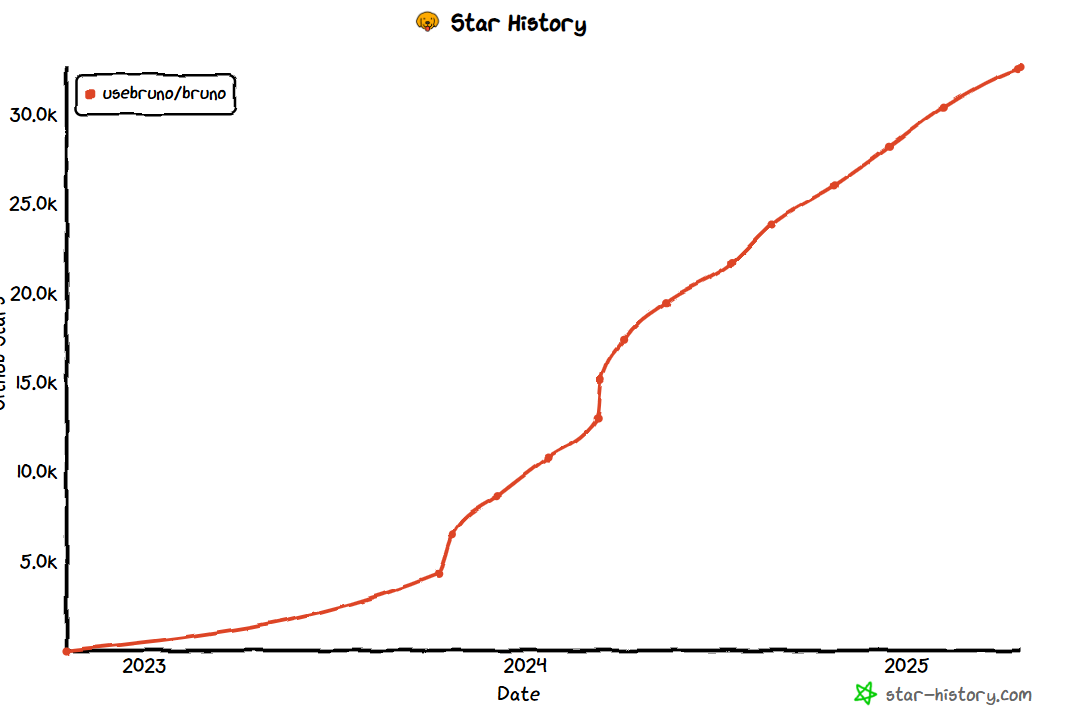
但作为一个商用工具，它的很多高级功能都依赖云端服务，且收费不菲。虽然免费的基础版本同样功能强大，能满足大部分使用需求，但随着工具本身和云端服务的整合以及更多向协作平台方向发展，基础版本也需要注册并登录云端服务后才可完全使用。

但这却限制了很多脱离互联网环境的应用需求，而且频繁的同步机制和连接稳定性要求也颇受诟病。

很多受Postman这些限制困扰的同学也一直在寻找和Postman类似的替代工具，本文我们向大家介绍一个API接口测试明日之星，开源接口测试工具 Bruno

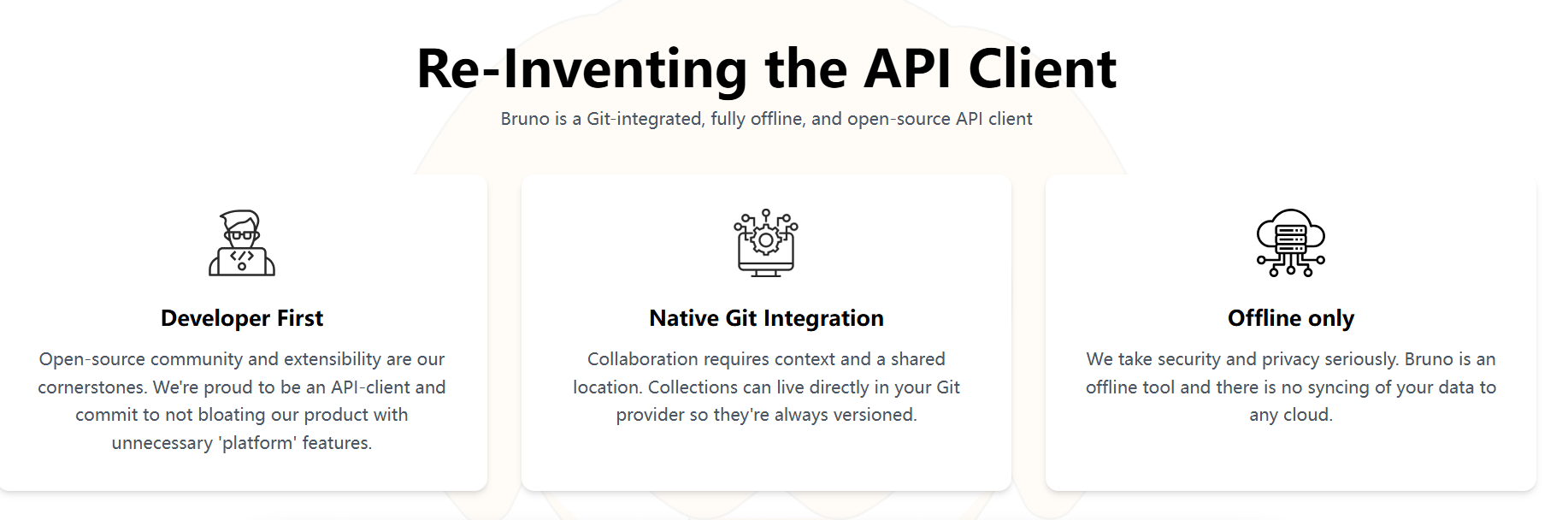


从github的star趋势可以看到，Bruno诞生不久，但从2024年开始，已经开始得到广泛关注，star数量一路飙升，潜力无限！



## Bruno简介

可以说 Bruno 从诞生之初针对的就是前述 **Postman** 的痛点。官方首页上强调的 Bruno 主要特性也重点说明了这几点：



1. 承诺开源和可扩展，并且专心做客户端，永不向“平台化”扩充产品。
2. 基于Git实现协作和版本管理
3. 离线使用，永不同步本地数据 😂😂😂

可以说，Bruno的出现，就是针对Postman作为一个客户端工具，过于强调平台化和云端功能的现状，提供的平替方案。

## Bruno的优势

Bruno在官方网站上，也是专门针对 Postman ，将Bruno和Postman进行了详细比对（当然一家之言，免不得自吹，大家辩证看待）网址如下： https://www.usebruno.com/compare/bruno-vs-postman

我整理了一下，主要包含如下几个维度：

#### 整体定位

| **Bruno** | **Postman** |
| --- | --- |
| 开源、本地优先、注重隐私 | 最初是 API 客户端，现已发展为包含云协作功能的 API 平台 |

#### 数据存储

| **Bruno** | **Postman** |
| --- | --- |
| 集合直接存储在本地文件系统，使用名为 Bru 的纯文本标记语言 | 主要存储在 Postman 云端，需要账户登录（虽然可以导出，但核心工作流依赖云） |

#### 开放源码

| **Bruno** | **Postman** |
| --- | --- |
| 完全开源 (MIT 许可证) | 部分开源，核心平台和云服务是专有的 |

#### 离线支持

| **Bruno** | **Postman** |
| --- | --- |
| 完全支持离线工作，因为所有数据都在本地 | 核心功能（尤其协作）严重依赖在线连接和账户登录 |

#### 协作方式

| **Bruno** | **Postman** |
| --- | --- |
| 通过 Git 或任何版本控制系统进行协作，利用其文件存储特性 | 通过 Postman 内置的云同步和团队协作功能（通常需要付费计划） |

#### CLI命令行

| **Bruno** | **Postman** |
| --- | --- |
| 提供名为 bru 的 CLI 工具，用于运行集合和测试 | 提供名为 Newman 的 CLI 工具，用于运行集合和集成到 CI/CD |

#### 脚本支持

| **Bruno** | **Postman** |
| --- | --- |
| 支持声明式脚本（Bru Lang）和 JavaScript 进行自动化测试和脚本编写 | 支持 JavaScript 进行请求前脚本、测试脚本编写 |

#### 交互界面

| **Bruno** | **Postman** |
| --- | --- |
| 界面和功能相对更简洁，专注于 API 请求和测试的核心任务 | 功能更全面，但也可能更复杂，涵盖了 API 设计、文档、监控等更多方面 |

#### 版本控制

| **Bruno** | **Postman** |
| --- | --- |
| 可直接利用 Git 进行精细的版本控制 | 内置版本控制功能，但与 Git 的集成和灵活性不同 |

总体来说，Bruno针对Postman的主要痛点，都给出了对应的应对处理，如果使用Postman受限或更希望在非互联网线环境下进行接口测试，可以说Bruno就是最佳选择

## Bruno的局限性

虽然Bruno已经受到了极大的关注，而且功能也足够丰富，但相比Postman这样老牌头部工具，功能上还是有一些局限性。这里总结起来，大致有如下几点：

#### 云端协作和同步

完全本地化和简洁、轻量，是Bruno重点突出的优势，但换一个角度来说，也是它主要的局限。

Bruno的团队协作以来Git来实现，对于熟悉Git使用的研发团队来说，当然足够强大，但如果团队对API的协作牵涉到混合团队时，比如包含客户、设计人员、产品经理等时，Git的使用门槛就相对更高了。Postman那样的基于云服务的团队功能则更易于接受，团队共享工作区、并可进行评论、共享。

#### 在API全生命周期管理上的欠缺

还是因为纯本地客户端的原因，Bruno很难像Postman那样实现针对API的全生命周期管理和支持，包括接口设计（OpenAPI、swagger等接口设计的导入）、交互式文档（API Documentation）、接口模拟（Mock Server）、健康监测（monitor）等，Postman都有对应的功能可以覆盖，而Bruno则无法涵盖如此丰富的场景。

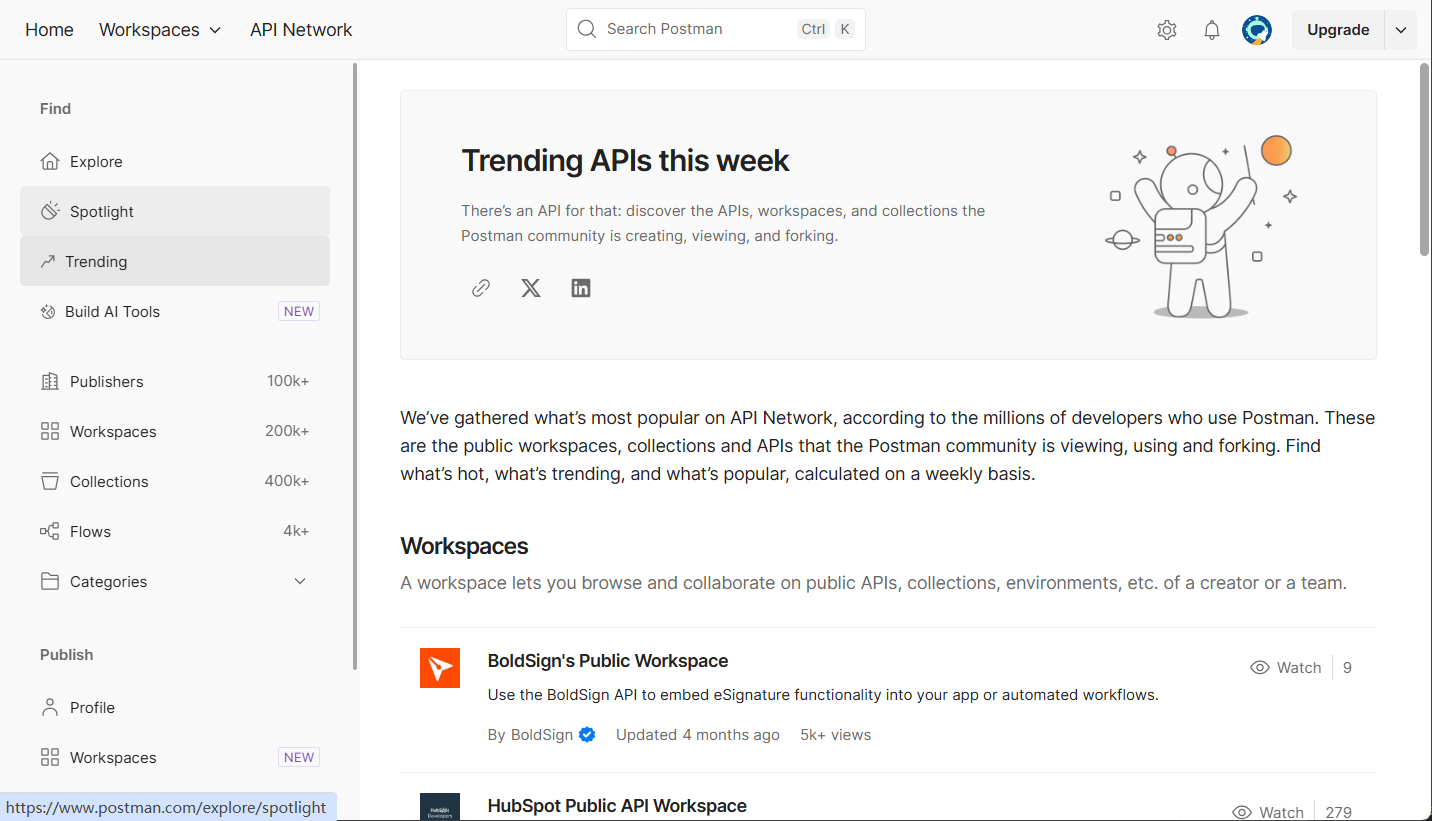
#### 接口类型支持不如Postman丰富

虽然Bruno目前已经支持restful和GraphQL接口，已经能够满足我们绝大多数的接口测试协议要求。但相比Postman还有MQTT这样的物联网协议支持，在协议扩展上，还有待继续补充。

#### 生态系统不如Postman成熟

* Postman 经过多年发展，已经围绕其平台构建了一个庞大的生态系统，包括大量的第三方工具集成（如很多第三方的 APM 工具、API 网关、CI/CD 工具等）都对Postman支持良好）。
* 经过多年积累，Postman社区繁荣，用户数量庞大，积累了海量的资源和教程。

Bruno 作为一个较新的工具，其原生集成和社区支持的插件相对较少



#### AI测试能力的支持

Postman通过内置的 postbot 助手，借助AI能力，可以实现接口测试用例的自动生成，优化建议等典型的AI增强。而随着AI应用的越来越广泛，对AI的支持也是Bruno还需要重点补充的部分。

## Bruno安装和应用基本示例

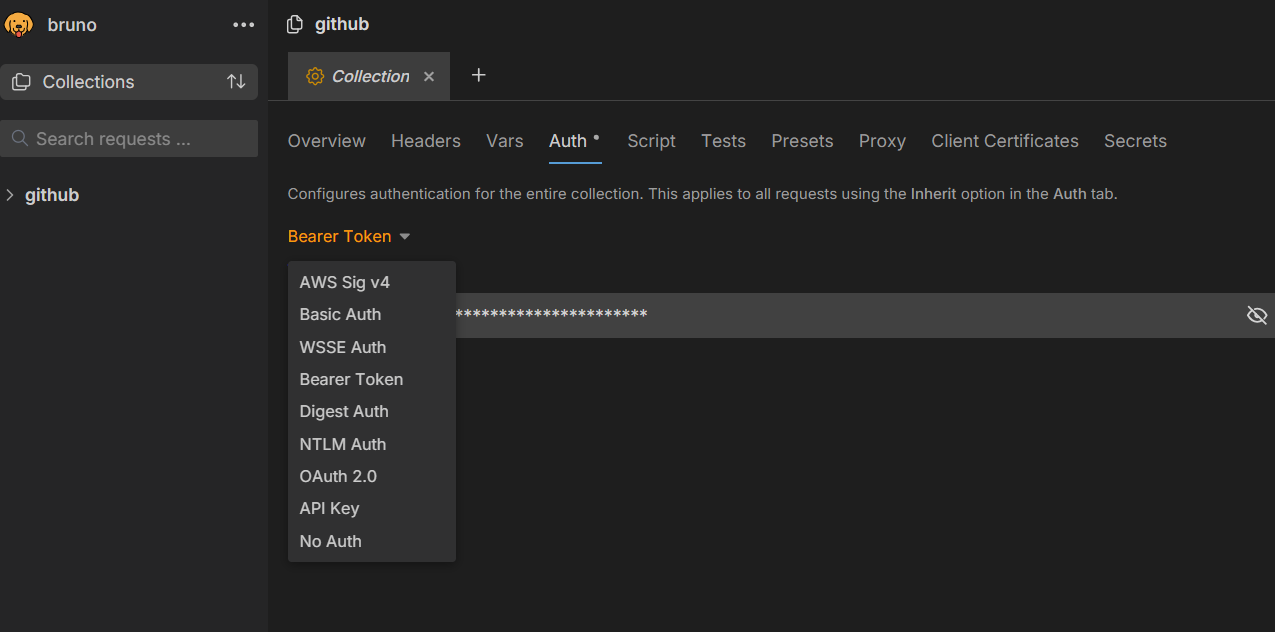
Bruno作为一个纯客户端，安装其实极为方便。访问官网，下载对应安装包即可(https://www.usebruno.com/downloads)。

win11下也可以利用内建的winget命令直接下载安装

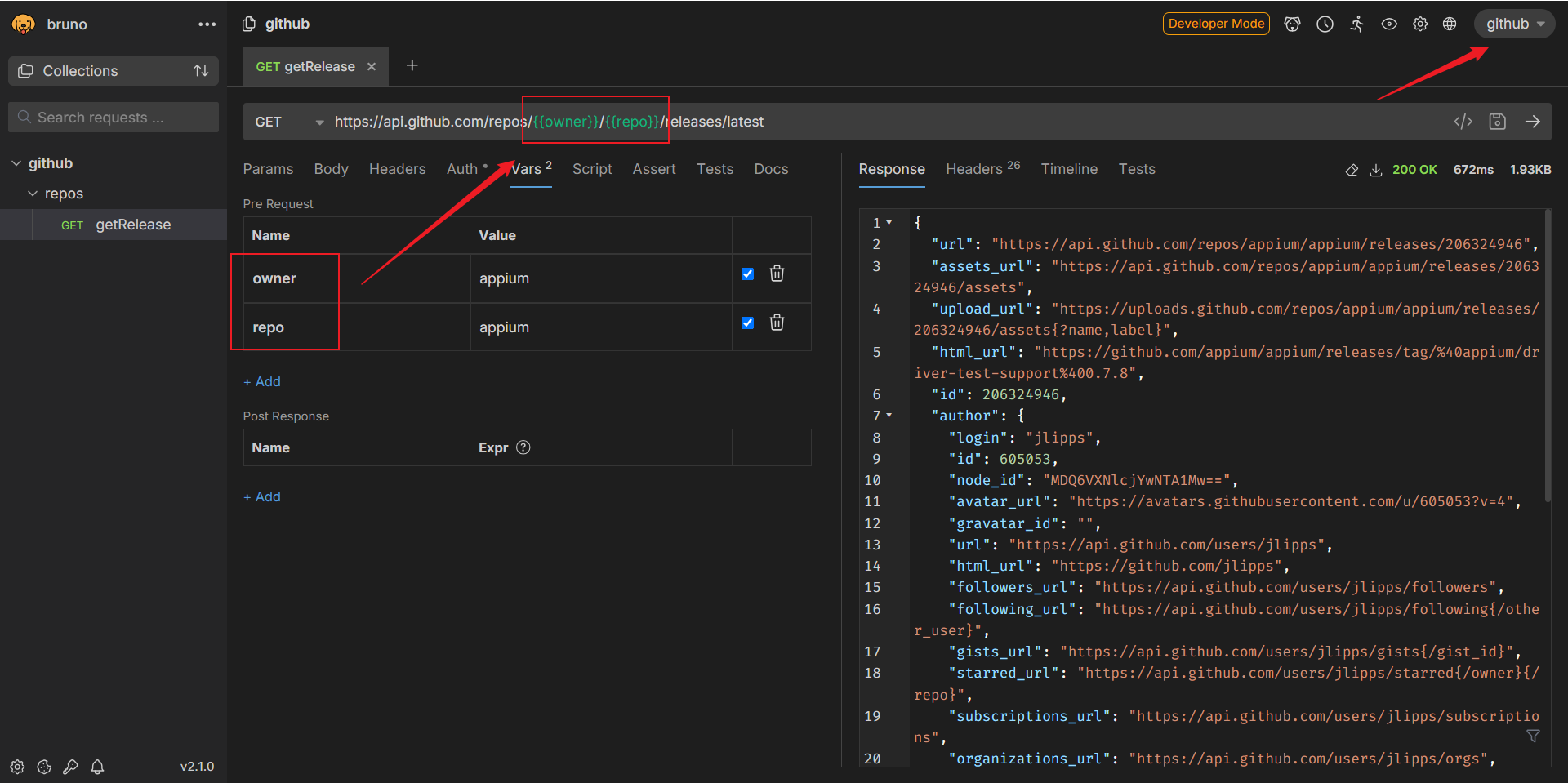
winget install bruno.bruno

安装后，桌面上会出一个bruno的小狗logo，类似Postman，双击就可打开Bruno客户端，界面其实和Postman颇为类似。下面我们以获取Github上一个开源项目的最近一次发布为例，看下Bruno中如何操作。

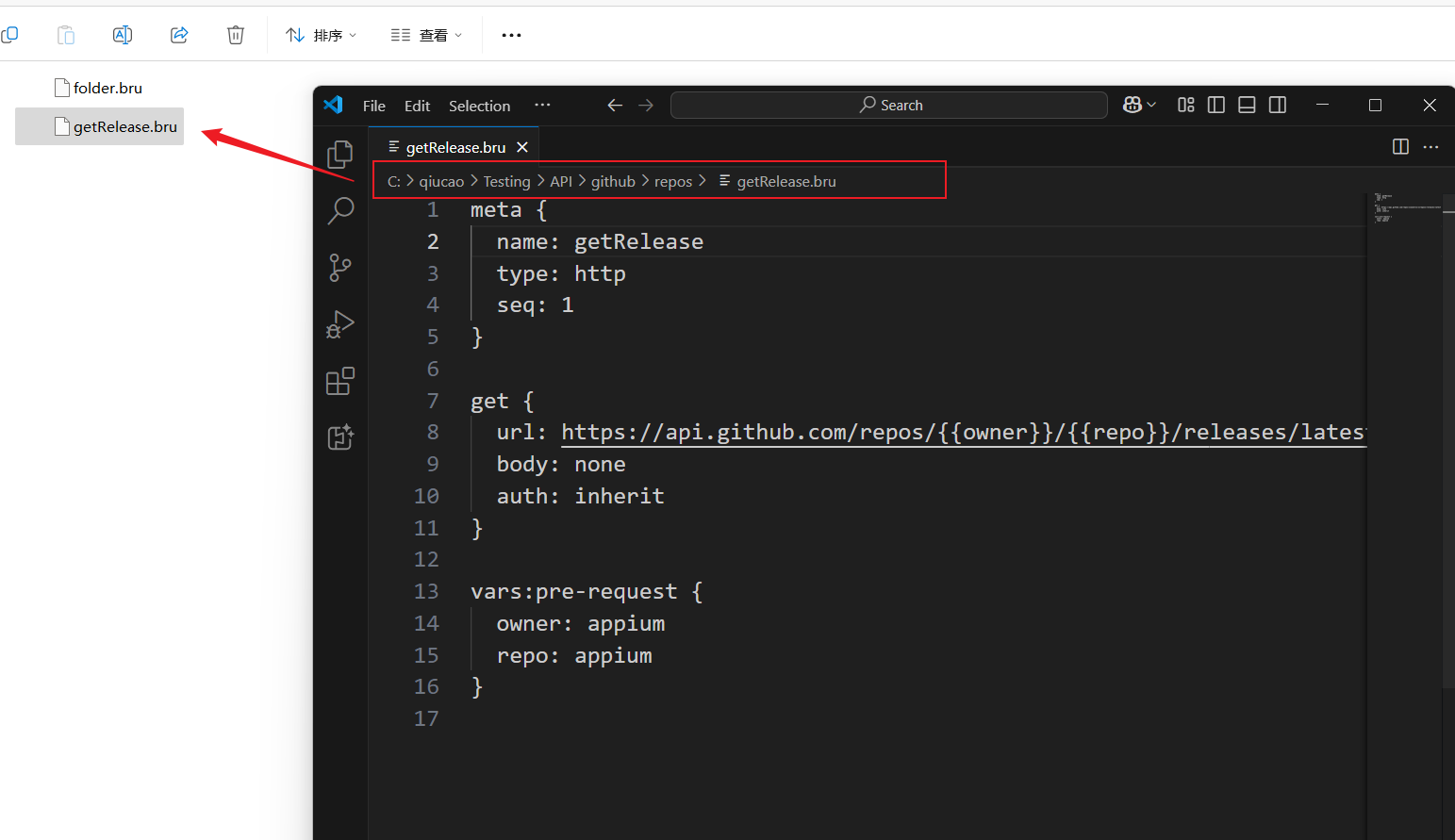
1. 首先建立一个collection，命名为 “github”, 并在其下创建一个目录，“repos”。 并可以在collection设置界面，添加github的访问Token。这里可以看到，Bruno支持的鉴权方式已经非常全面。



1. 在 “repos” 目录下，新建一个Get请求，“getRelease”, 根据Github定义，对应请求接口 “https://api.github.com/repos/{{owner}}/{{repo}}/releases/latest”， 这里我们使用了两个变量，和Postman类似，这里设置的变量可以通过{{var}} 来引用。 测试一下，成功获取了appium项目的最近一次发布信息



作为一个本地客户端，这里Bruno所有的请求和配置信息，其实都保存在本地，.bru文件中进行存储。



以上就是关于Bruno的几个概要介绍，对于不希望受Postman在线限制的同学，Bruno就是最佳替代。

欢迎继续关注秋草的后续博客文章，会陆续带来更多关于Bruno的深度使用总结和分享！