

OpenDigger API 使用指南

输入一个 GitHub 仓库名，就能获取它的活跃度、贡献者、Issue、PR 等统计数据。

怎么用

最简单的方法：直接访问在线数据

OpenDigger 把它采集到的所有数据都放在了一个公开的网址上，不需要注册、不需要 Token，直接访问就行，**但是他没有爬取到所有人的所有仓库。**

格式：

```
https://oss.open-digger.cn/github/{owner}/{repo}/{指标名}.json
```

例子：

```
https://oss.open-digger.cn/github/apache/echarts/activity.json  
https://oss.open-digger.cn/github/apache/echarts/openrank.json  
https://oss.open-digger.cn/github/apache/echarts/stars.json
```

用浏览器打开就能看到 JSON 数据。

能获取什么数据

基础指标

- `activity.json` - 活跃度（每月有多活跃）
- `openrank.json` - 影响力指数
- `stars.json` - Star 数变化
- `participants.json` - 参与者数量

Issue 相关

- `issues_new.json` - 新增 Issue 数量
- `issues_closed.json` - 关闭的 Issue 数量
- `issue_response_time.json` - Issue 响应时间
- `issue_resolution_duration.json` - Issue 解决时长
- `issue_age.json` - Issue 存活时间

PR 相关

- `change_requests_accepted.json` - 被接受的 PR 数量
- `change_requests_declined.json` - 被拒绝的 PR 数量
- `change_request_response_time.json` - PR 响应时间

- `change_request_resolution_duration.json` - PR 处理时长

贡献者相关

- `new_contributors.json` - 新增贡献者
- `bus_factor.json` - 总线因子（核心贡献者集中度）
- `inactive_contributors.json` - 不活跃的贡献者

其他

- `technical_fork.json` - Fork 数量
- `code_change_commits.json` - 代码提交数

限制

不能获取的数据

OpenDigger **只提供统计数据**, 不提供原始文本内容 :

不能获取 :

- Issue 的标题和内容
- Issue 的评论
- PR 的描述和评论
- Commit 的详细信息
- 用户的个人信息

能获取 :

- Issue 的数量统计
- Issue 的响应时间统计
- PR 的接受/拒绝数量
- 贡献者数量统计
- 活跃度趋势

OpenDigger 并没有爬取所有人的所有仓库，会造成遗漏（比如他就没搜到我们的信息）

如果需要 Issue 或者其他的文本信息怎么办？

使用 GitHub 官方 API :

- 文档 : <https://docs.github.com/en/rest>
- 需要 GitHub Token
- 有访问频率限制
- 可以实现的，但是不能大规模爬。比如一次爬一个可以实现。但是一次爬几百个那就歇菜了。

如果需要爬取被遗漏的人的仓库怎么办？

使用 GitHub 官方 API :

- 需要 GitHub Token（很多个，便于轮换），然后重新写爬虫的脚本。
- 有访问频率限制

- 也是可以实现的，但是不能大规模爬，可能还会有速率限制。并且一个人的账户如果啥都没有，爬下来价值也不大。

数据格式

所有数据都是 JSON 格式，通常是时间序列（基于这个性质我们可以考虑做时间序列分析）：

```
{  
    "2020-01": 123.45,  
    "2020-02": 234.56,  
    "2020-03": 345.67  
}
```

键是日期（年-月），值是该月的统计数据。

快速开始

1. 浏览器直接查看

打开浏览器，访问：

```
https://oss.open-digger.cn/github/microsoft/vscode/activity.json
```

2. 使用 Python 脚本

运行脚本：

```
python opendigger_api.py
```

输入仓库名（如 microsoft/vscode），自动获取所有数据并保存为 Excel。

总结

适合用 OpenDigger 做什么？

1. **看项目是不是还活跃**：通过活跃度、提交数、新增贡献者等指标，判断项目是上升期还是夕阳红，也可以做预测。
2. **对比不同项目**：比如想在同类型的 A 和 B 之间选一个好的，可以对比它们的活跃度、响应速度、贡献者数量
3. **评估项目健康度**：Issue 响应快不快、PR 接受率高不高、核心贡献者会不会太集中（总线因子）

不适合用 OpenDigger 做什么？

1. **分析 Issue 内容**：比如想知道用户最常反馈什么问题、有哪些功能需求，OpenDigger 没有这些文本数据（但是可以针对完善）
2. **情感分析**：想分析社区氛围好不好、用户评价如何，得用 GitHub API 爬评论

3. 深入研究具体问题：比如想看某个 Bug 的讨论过程、某个 PR 为什么被拒绝，得去 GitHub 页面看