

## turbolinks/turbolinks 项目分析：

Turbolinks 是源自 Ruby on Rails 的 Web 加载优化方案。简单来说就是当用户点击链接时，并不真实的跳转网页，而是通过 ajax 读取目标页的内容，然后替换当前页。

### 一、项目的基本背景和发展历程介绍

#### 1.技术类型

主要使用了 TypeScript 和 JavaScript 进行开发。

#### 2.版本发布历史

- 2017.05.03 发布 v5.0.1
- 2017.05.03 发布 v5.0.2
- 2017.05.05 发布 v5.0.3
- 2018.01.09 发布 v5.1.0
- 2018.01.09 发布 v5.1.1
- 2018.07.28 发布 v5.2.0.beta1
- 2018.08.20 发布 v5.2.0
- 2019.03.12 发布 v5.3.0-beta.1

#### 3.主要贡献者的构成（国家、区域和组织等）

Sam Stephenson：贡献了 171 条 commits

Jeffrey Hardy：Ontario, Canada，贡献了 48 条 commits

Javan Makhmali：Ann Arbor, MI，贡献了 30 条 commits

Dom Christie：Los Angeles，贡献了 14 条 commits

### 三、洞察项目被归档的可能原因

#### 1. 阅读分析项目的相关信息：

##### (1) 主页、主要贡献者发表的相关技术博客

Sam Stephenson 近几年发表了 bats (Bash Automated Testing System)、eco (Embedded CoffeeScript templates)、ruby-ejs 项目。

Jeffrey Hardy 近几年发表了 rails、spicy-proton、activeresource 等项目。

Javan Makhmali 近几年发表了 input-inspector、details-element-polyfill、marcel 等项目。

Dom Christie 近几年发表了 mixmark-io/turndown 、bgrins/ExpandingTextareas 等项目。

##### (2) Issue 和 PR 中的相关讨论

Issue 和 PR 的讨论中，大概有五分之一是指出项目的 bug 和 enhancement 问题，其余基本都是提出问题，并且绝大部分已被解决。

##### (3) README 文件，贡献者指南，Code of Conduct 及其他可能有的相关文档

README 文件中主要涉及了项目特色、具体安装方法、使用说明、参考 API、贡献者部分。

Code of Conduct 文件中涉及 Pledge、Standards、Responsibilities、Scope、Enforcement 、 Project Maintainers 部分。

#### 2. 结合一和二中得到的信息和分析结果，尝试总结项目可能的归档原因

根据二中得到的结果，可以看出自 2016 年创建项目以来，每个月新增 star 个数呈先增长而后下降的趋势，尤其 2021 年每月新增 star 个数极少，后期趋近于 0；而每月新增 fork 个数的均值逐年下降，在 2021 年下降最明显；每月新增 issue 个数从 2016 年平均 15 个以上逐年下降，直至 2021 年平均每月只有一两个；每月关闭 issue 个数逐年下降，且后三年数量极少；每月打开 pr 个数也在逐年减少，2021 年几乎为 0；每月合入 pr 个数在前三年先上升，而后三年逐渐下降直至趋于 0。

根据以上数据，可以看出，随着项目的不断开发与完善，该项目的新增 star 与 fork 逐渐下降，代表被关注度逐渐减弱。且 issue 越来越少被提出，也越来越少被解决，而开发者在后两年尤其是 2021 年也明显表现出对继续维护项目的倦怠。

随着更多新技术的开发与实现，目前的框架模型可能已经不足以满足用户需求，或许已经出现新的框架可以取代它，从而使得关注度的逐年下降。而关注度的下降更加熄灭了开发者继续开发和维护的热情，因而导致该项目被归档。

### 3.结合你搜集到的信息，尝试分析项目归档后可能产生的影响（对开发者和用户）

对开发者而言，项目归档意味着不需要开发者再投入大量的时间精力去修改和维护项目，因此可以更高效地投入其他工作或是其他项目的开发，但同时也因此无法接受更多用户的反馈，失去了继续完善优化该项目的机会。

对用户而言，项目归档使得用户无法再与开发者进行有效沟通，如果产生新的问题也无法得到解决，而且由于开发者不再对项目进行维护可能会令软件出现越来越多的漏洞，使得用户无法长期使用该项目。

### 4.表述你对开源项目如何可持续发展的理解

近来，关于如何维持开源项目的可持续发展问题，有越来越多的问题亟待解决。

#### （1）资源问题

开源社区无处不在，但普遍缺乏经济资源和人力资源。人们重视实体基础设施的投资，但却忽视了数字基础设施的建设。像实体基础设施一样，数字基础设施也需要不断的定期维护，很多比较有价值的开源软件无法得到商业模式的支持，也缺乏任何形式的机构支持。从今天开源社区资源获取的情况来看，大部分资源被投入到少数几个热门的项目上，其他的一些项目受关注程度和获取的资源支持相对较少，这种现象也会导致一些问题的出现，例如一些基础软件和小众软件的生存空间可能逐渐被压缩掉。

因此为了更多开源项目的可持续发展，应该有更多的中小型开源项目被重视起来，多支持多鼓励这些项目的开发与建设比如支持小项目商业化。

#### （2）项目维护及管理

据统计，很大一部分开源软件在开发出来之后得不到必要的维护，很多项目的开发人员只有少数一两个，甚至某些项目完全由志愿者开发并维护，因此常出现出现问题后没有人解决的严重后果。且很多项目的主要开发人员尤其是核心技术人员可能会在开发中途退出，严重影响了项目进展及日后维护工作。

### (3) 成本与回报问题

开源文化中普遍提倡奉献精神，往往认为涉及金钱会剥削开源工作的资源精神，违背很多人参与开源工作的初衷。但实际上，开源工作者也会面临很多资源需求，在既无法获得报酬而开源工作又与自己的现实生活和工作发生冲突时，许多开源工作者被迫选择放弃或退出。

对于开源可持续发展，可以通过改进众筹的方式增加项目收入，以此回馈开源贡献者。

分工：

代码——王柳苏（51215903112）、王心悦（51215903104）

文档——马悦祺（51215903110）