Mining Software Repositories (MSR)

John Businge
(john.businge@uantwerpen.be)
Henrique Rocha

Universiteit
Antwerpen

References

The Road Ahead for Mining Software Repositories

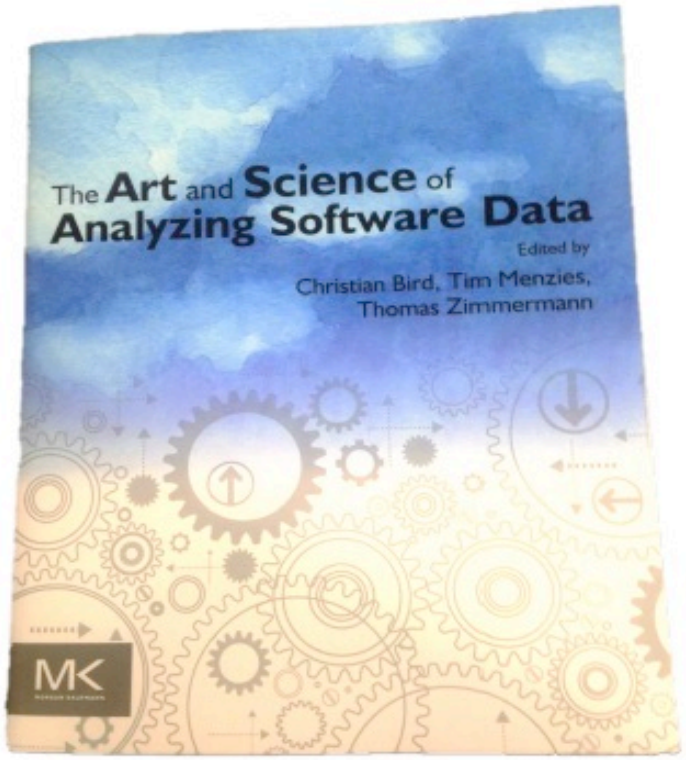
Ahmed E. Hassan
Software Analysis and Intelligence Lab (SAIL)
School of Computing, Queen's University, Canada
ahmed@cs.queensu.ca

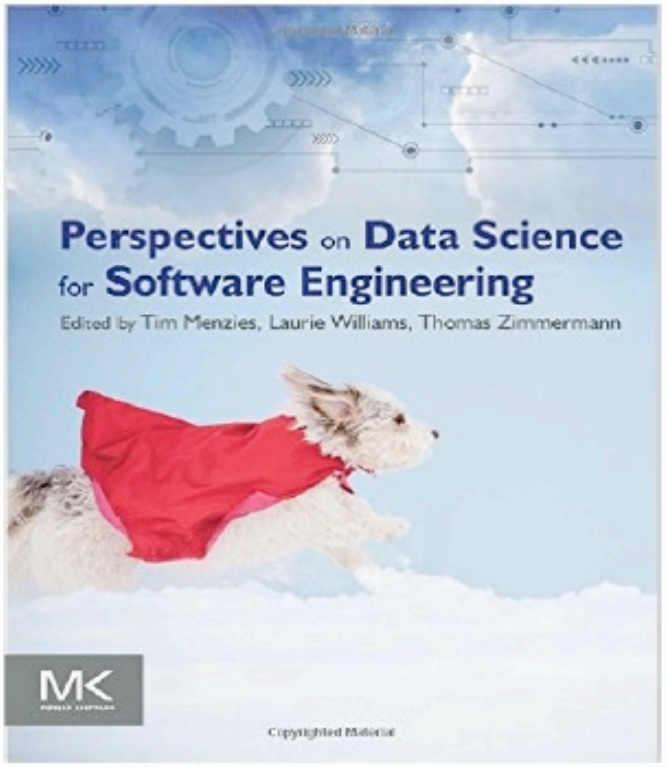
Software Intelligence:
The Future of Mining Software Engineering Data

Ahmed E. Hassan
School of Computing
Queen's University
Kingston, ON, Canada
ahmed@cs.queensu.ca

Tao Xie
Department of Computer Science
North Carolina State University
Raleigh, NC, USA
xie@csc.ncsu.edu

More References





Acknowledgement

Ahmed E. Hassan
Queen's University
www.cs.queensu.ca/~ahmed
ahmed@cs.queensu.ca

Tao Xie
North Carolina State University
www.csc.ncsu.edu/faculty/xie
xie@csc.ncsu.edu

Bram Adams
Polytechnique Montréal, Canada
http://mcis.polymtl.ca/index.html
lab.mcis@gmail.com

With updates by **John Businge** from University of Antwerp, Belgium

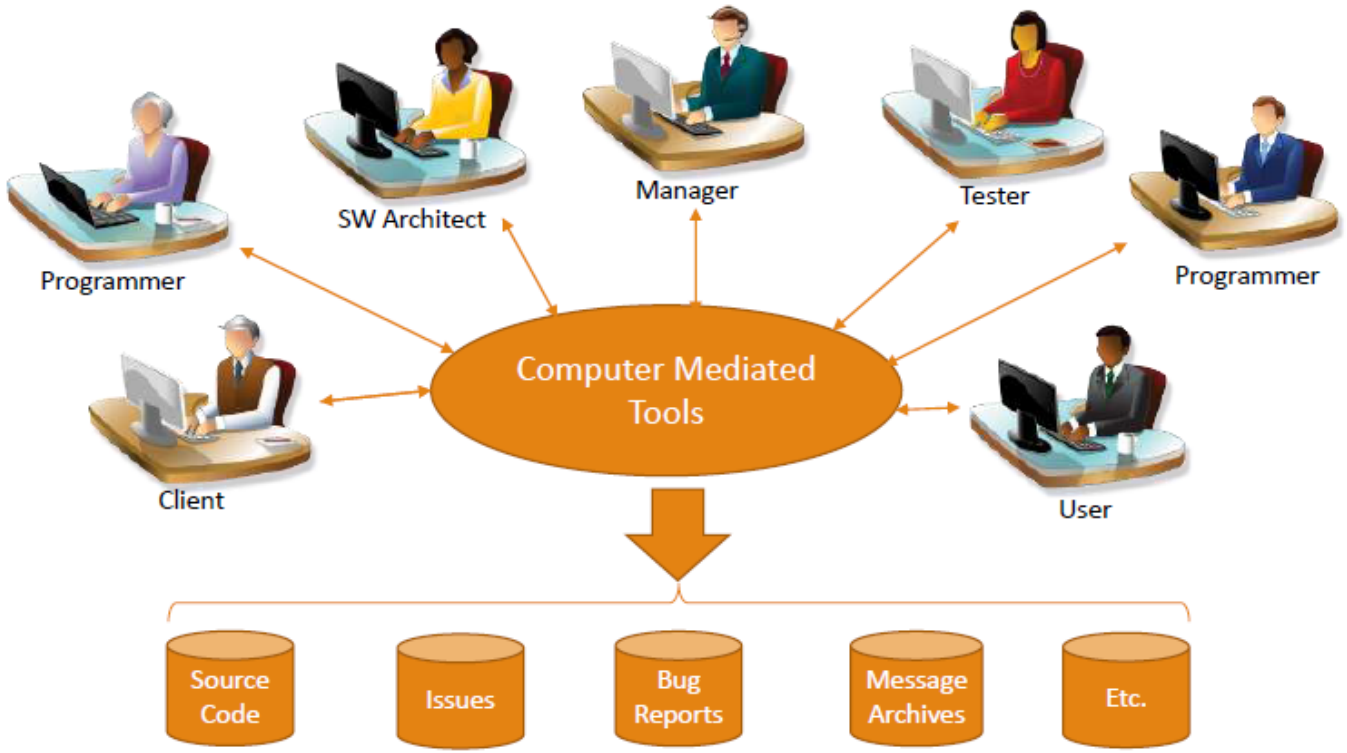
Lecture Goals

- **Learn about:**
  - Classic and notable research and researchers in mining software engineering (SE) data
  - Data mining and data processing techniques and how to apply them to SE data
  - Risks in using SE data due to e.g., noise
- **After the lecture, you should be able to:**
  - Retrieve SE data
  - Prepare SE data for analysis
  - Mine interesting information from SE data

Why mine SE data?

- **SE data can be used to:**
  - Gain empirically-based understanding of software development
  - Predict, plan, and understand various aspects of a project
  - Support future development and project management activities

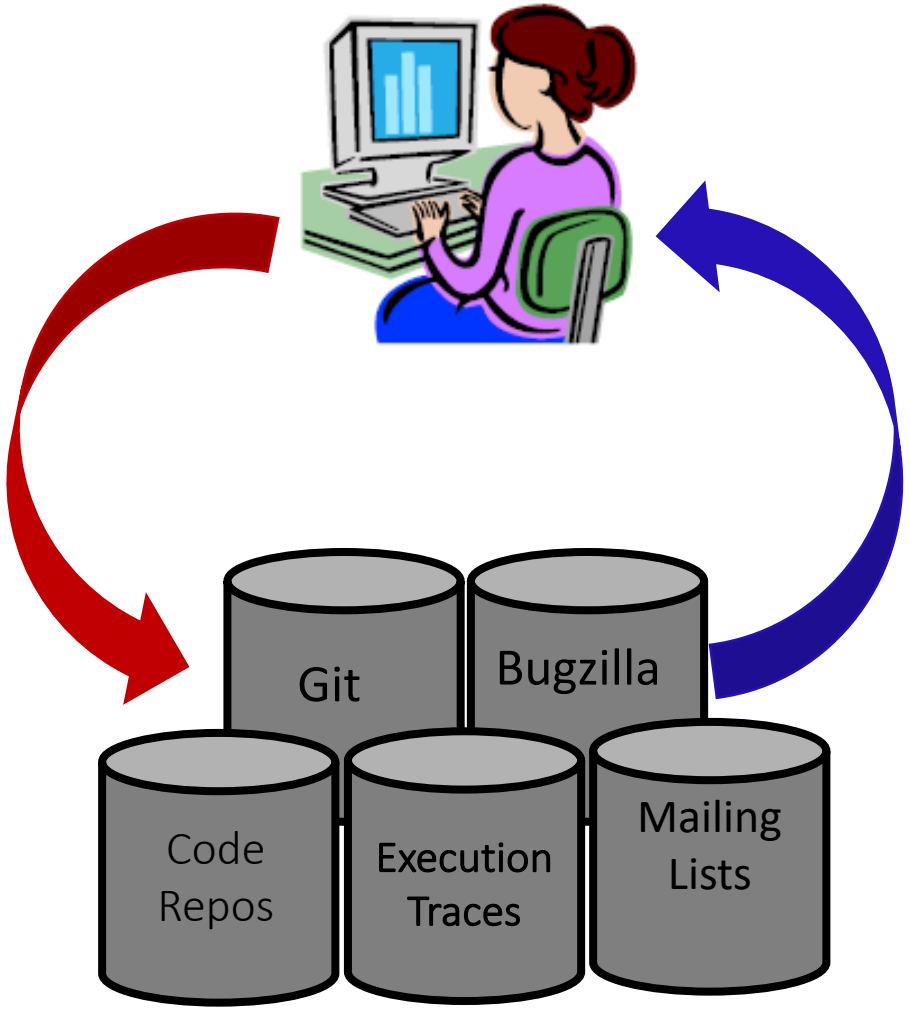
How is SE Data generated?



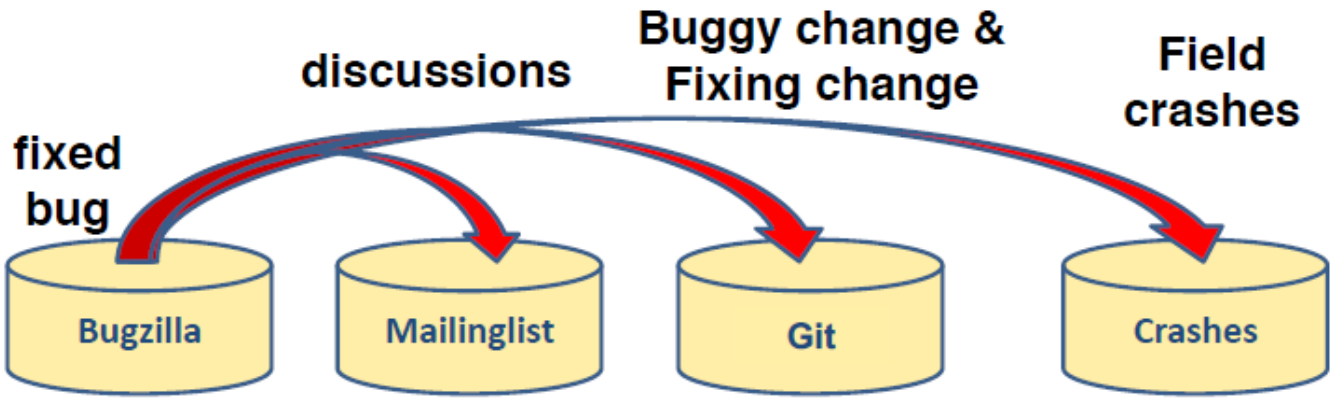
Current and historical artifacts and interactions are registered in software repositories

What is MSR?

- Transforming static record keeping SE into active data
- Making SE data actionable by uncovering patterns and trends



MSR researchers analyze and cross-link repositories



New bug report
Estimate fix effort
Suggest experts to fix the bugs!

What can we learn from SE data?

- Let us look at a sample of notable findings from MSR studies.

MSR studies – Bugs (1)

Using imports to predict Bugs

- **MSR**: bugs and imports
- Trained models of No. of **bugs** vs No. of **imports**

[Schröter et al. 06]

Results

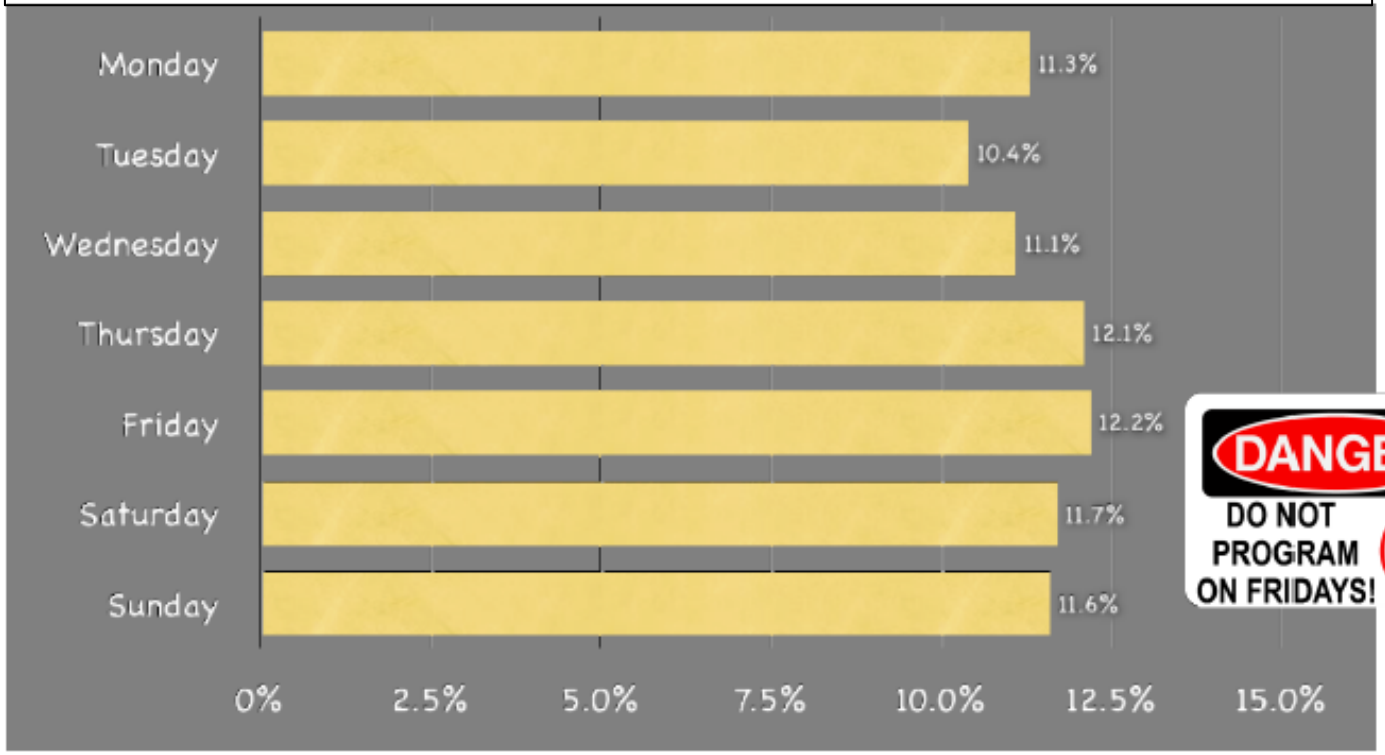
71% of files that import compiler packages, had to be fixed later on.

```
import org.eclipse.jdt.internal.compiler.lookup.*;
import org.eclipse.jdt.internal.compiler.*;
import org.eclipse.jdt.internal.compiler.ast.*;
import org.eclipse.jdt.internal.compiler.util.*;
...
import org.eclipse.pde.core.*;
import org.eclipse.jface.wizard.*;
import org.eclipse.ui.*;
```

14% of all files that import ui packages, had to be fixed later on.

MSR studies – Bugs (2)

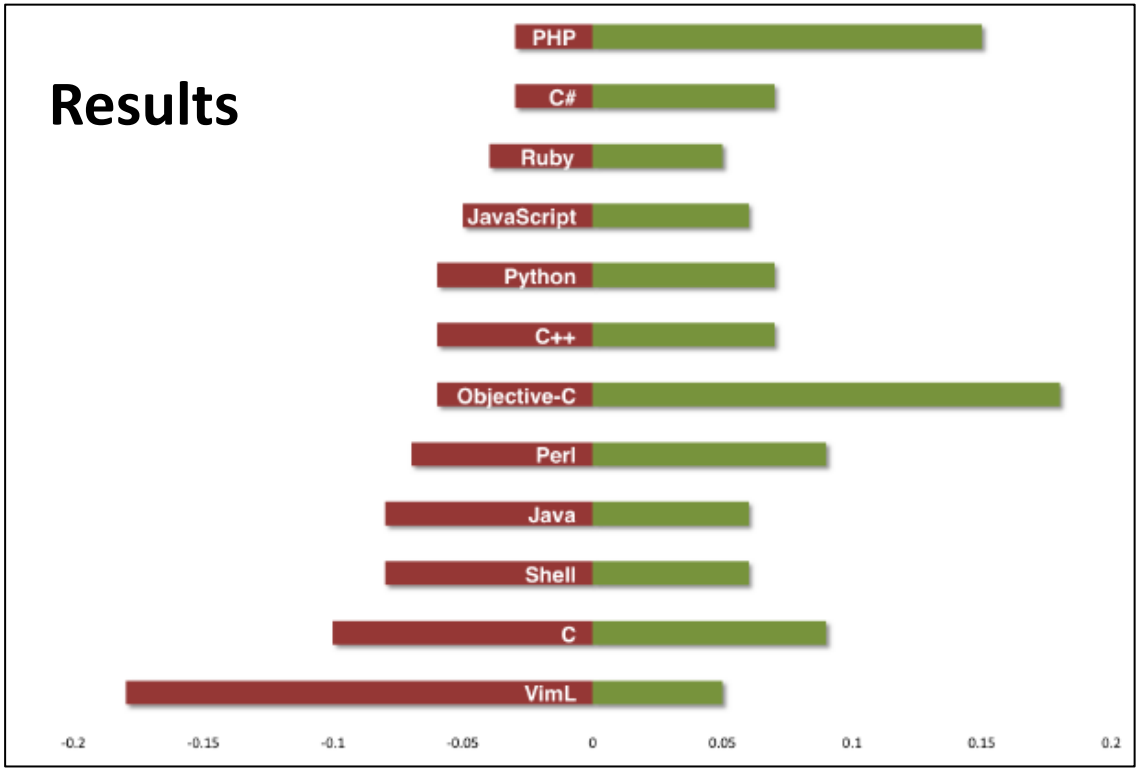
- **MSR**: CVS and Bugzilla (bug database) for the Eclipse framework
- **Results**: Fix-inducing changes showed a distinct pattern w.r.t days of the week.



Percentage of bug-introducing changes for eclipse [Zimmermann et al. 05]

MSR studies – Sentiment Analysis (1)

MSR: Mined commit logs, written in various programming languages and extracted phrases related to **Anger** and **Joy**



Anger vs. **Joy**

How they stack up?

- **PHP**, **Object-C**, and **C#** were net positive
- **Java**, **Shell**, and **C** were fairly even
- **VimL** is just bad news.

[Doll and Grigorik, 2012]

MSR studies – Sentiment Analysis (2)

Ortu et al., MSR-2015

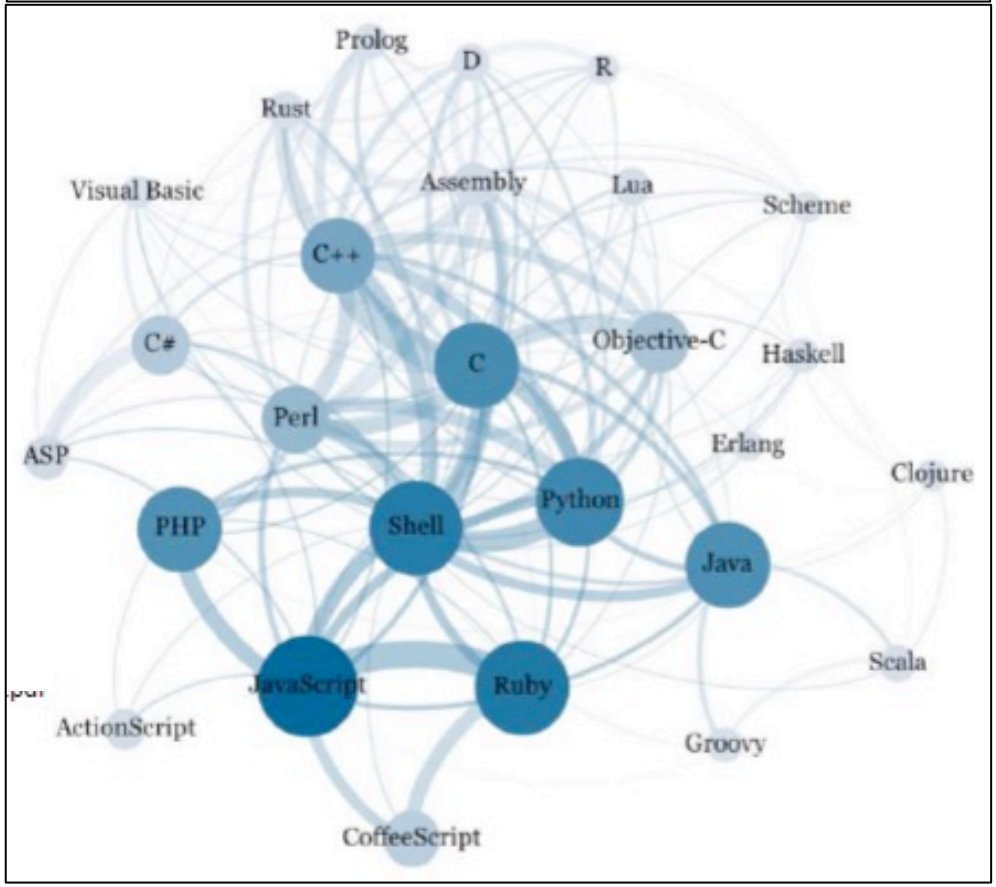
Are Bullies more Productive? Empirical Study of Affectiveness vs. Issue Fixing Time

Marco Ortu*, Bram Adams ‡, Giuseppe Destefanis†, Parastou Tourani ‡, Michele Marchesi * Roberto Tonelli *
*DIEE, University of Cagliari, Italy, {marco.ortu,michele,roberto.tonelli}@diee.unica.it
†CRIM, Computer Research Institute of Montreal, Canada, {giuseppe.destefanis}@crim.ca
‡École Polytechnique de Montréal, Canada, {bram.adams,parastou.tourani}@polymtl.ca

- **MSR**: Mined more that **560K Jira comments** as well as the **issue fixing times.**
- **Results**: They found that the happier developers (expressing emotions such as **JOY** and **LOVE** in their comments) had shorter the issue fixing times.

MSR studies – Programming languages

Visualising the GitHub programming language usage correlation dataset



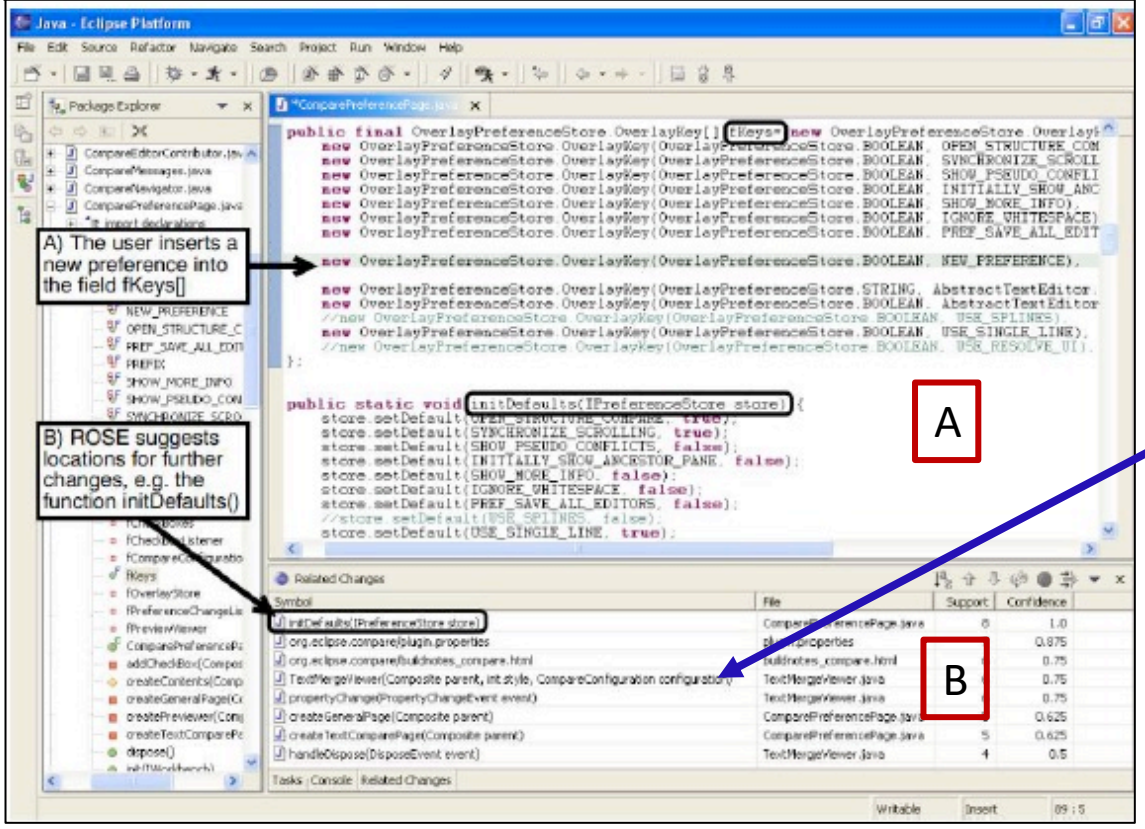
- A **Ruby** programmers **is very likely to know JavaScript**, while a **Perl** programmer is not
- **Java** is a popular programming language but stands primarily alone

https://github.com/mjwillson/ProgLangVisualise

MSR studies – Changes by programmers

MSR to uncover programming language relations

[Zimmermann et al., 2005]



After the programmer has made some changes to the source (above), **ROSE** suggests locations where further changes were made

ROSE tool guides the programmer by:

- Suggestinng and predicting likely changes
- Preventing errors due to imcoplete changes

Where can we mine SE Data?

Sept 2020:
100 Millinon repositories
56 Million Users

April 2019
28 Millinon repositories
10 Million Users

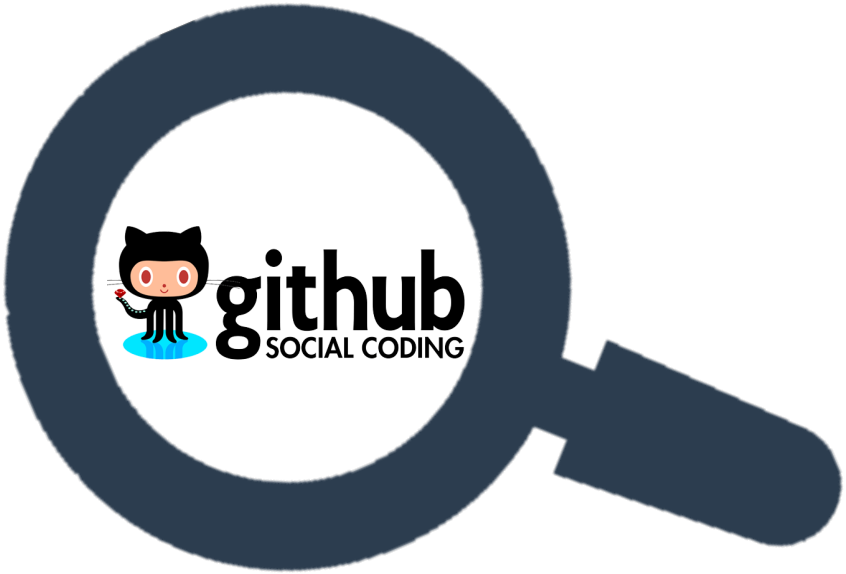
Libraries.io

Travis CI

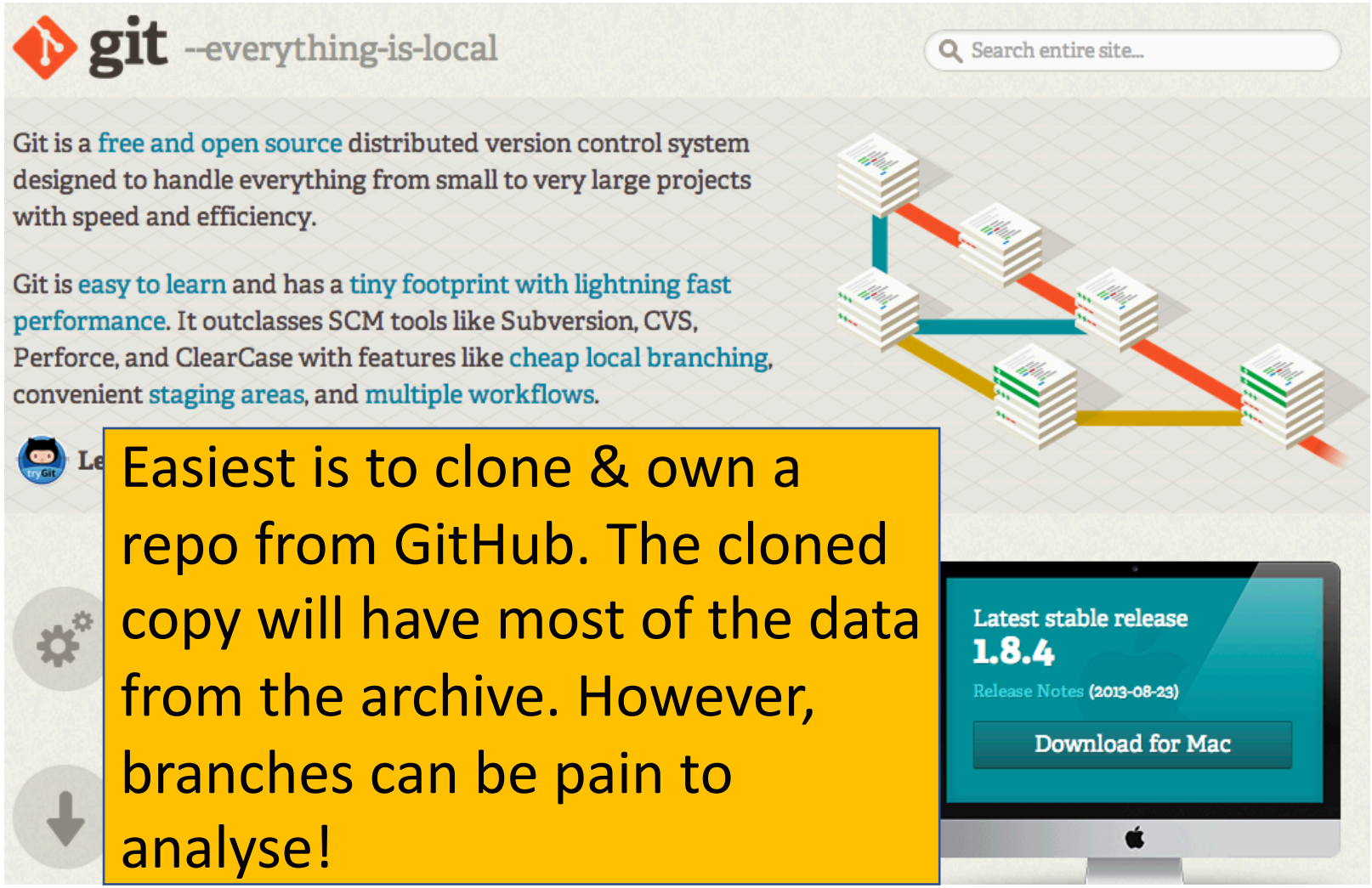
stackoverflow

How can we mine SE Data?

All the mentioned platforms and many more have got REST APIs

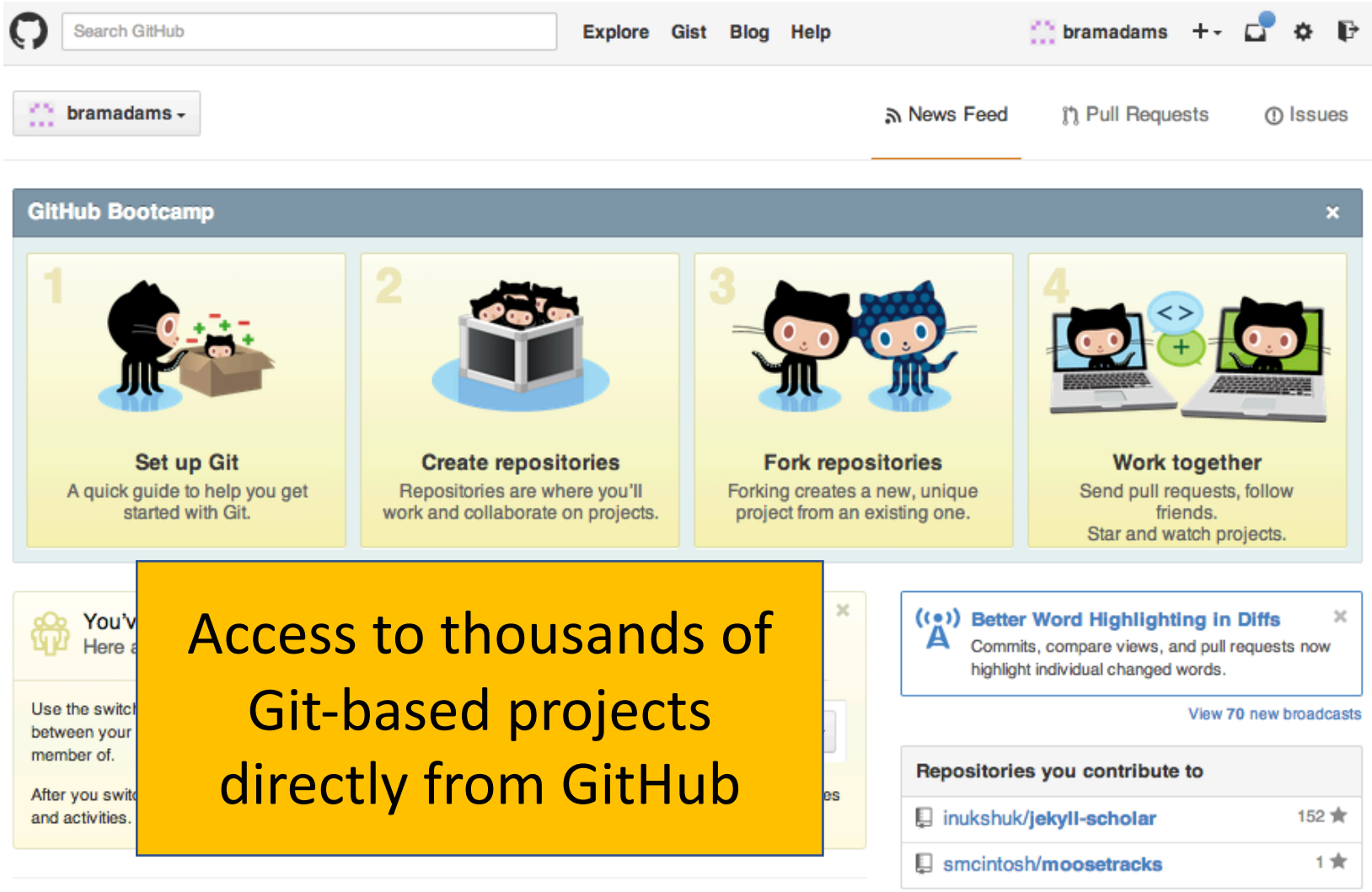


How can we mine GitHub Data?



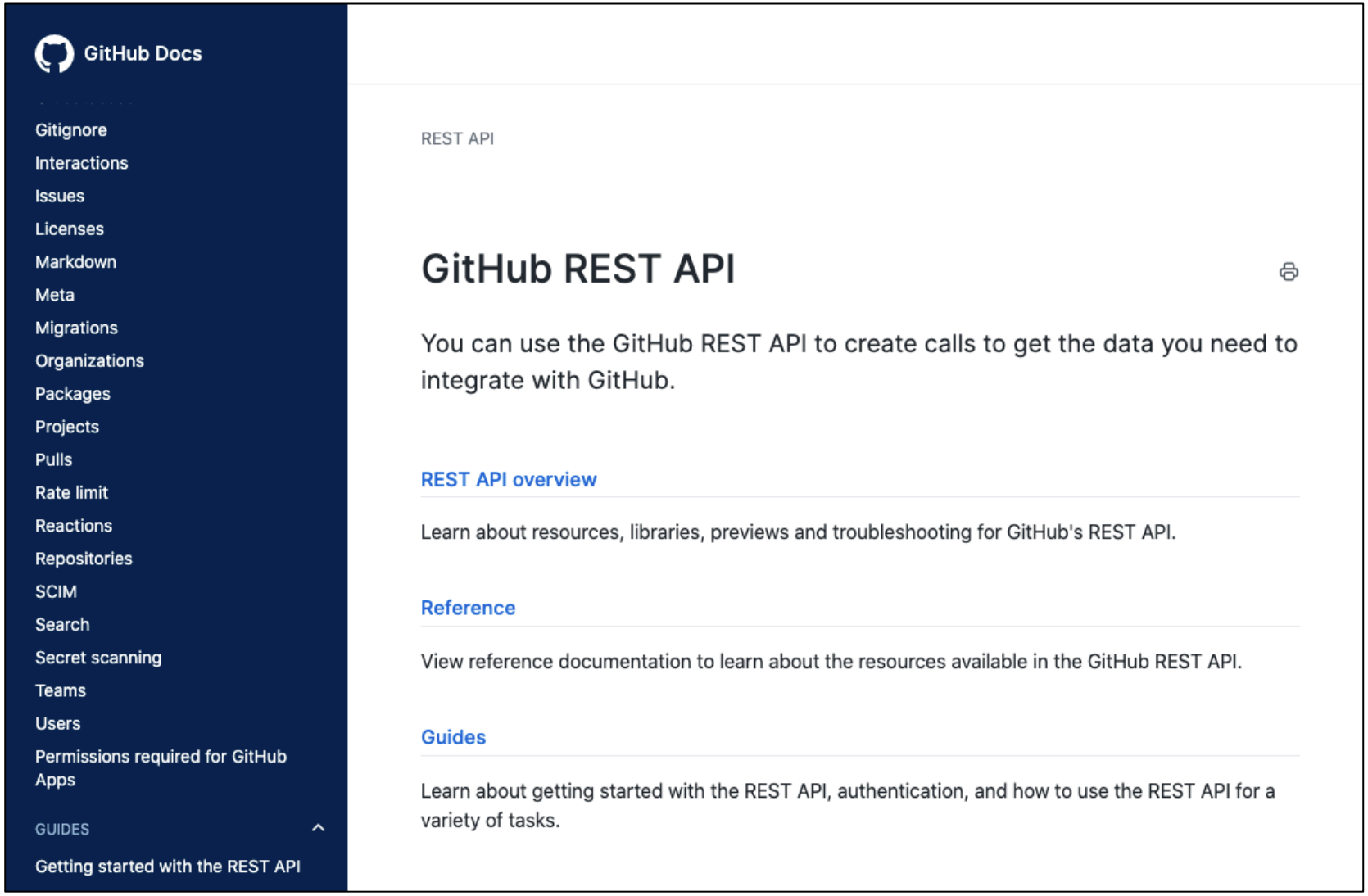
Easiest is to clone & own a repo from GitHub. The cloned copy will have most of the data from the archive. However, branches can be pain to analyse!

How can we mine GitHub Data?

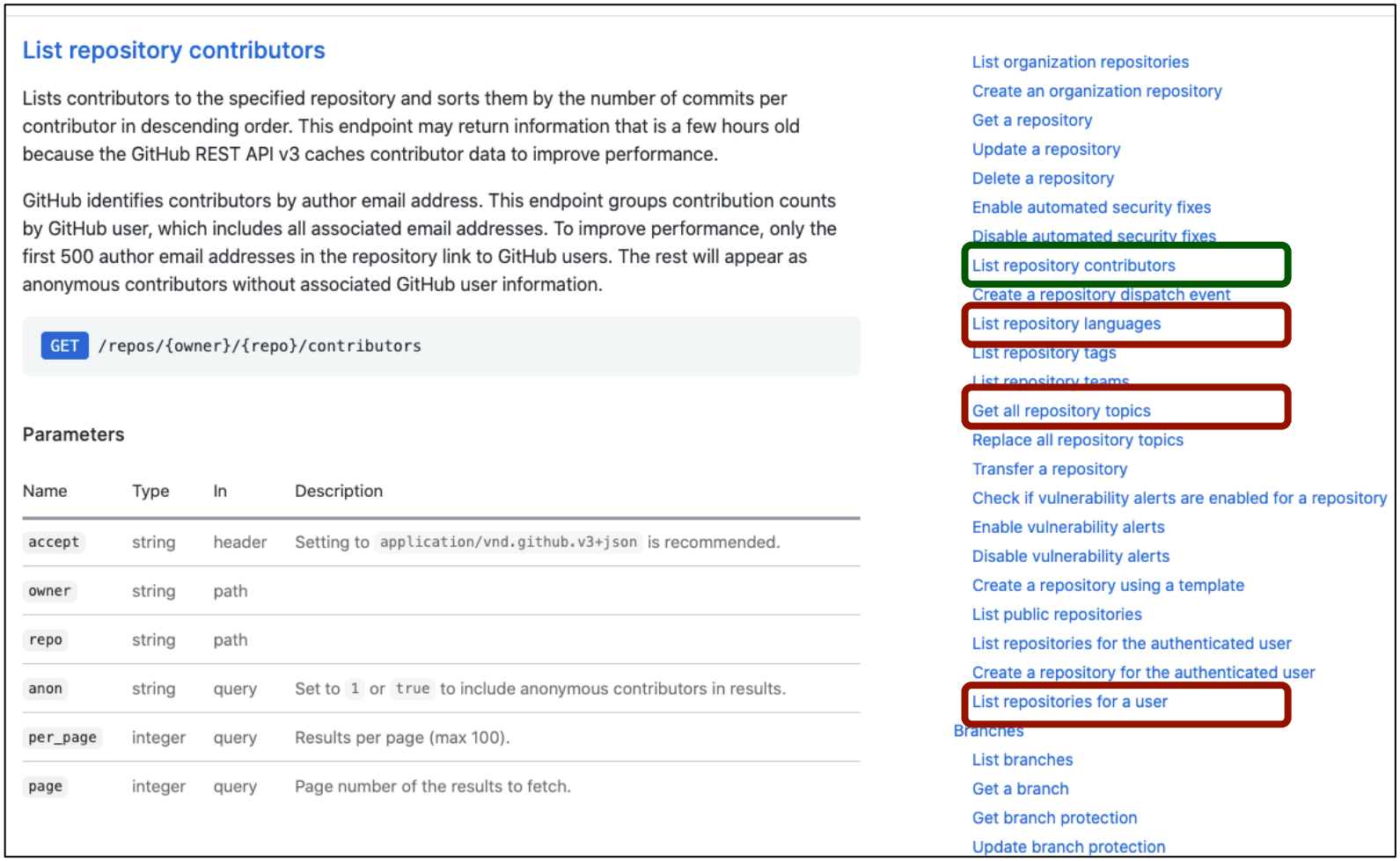


Access to thousands of Git-based projects directly from GitHub

How can we mine GitHub Data?



How can we mine GitHub Data?



List repository contributors

Lists contributors to the specified repository and sorts them by the number of commits per contributor in descending order. This endpoint may return information that is a few hours old because the GitHub REST API v3 caches contributor data to improve performance.

GitHub identifies contributors by author email address. This endpoint groups contribution counts by GitHub user, which includes all associated email addresses. To improve performance, only the first 500 author email addresses in the repository link to GitHub users. The rest will appear as anonymous contributors without associated GitHub user information.

`GET` `/repos/{owner}/{repo}/contributors`

Parameters

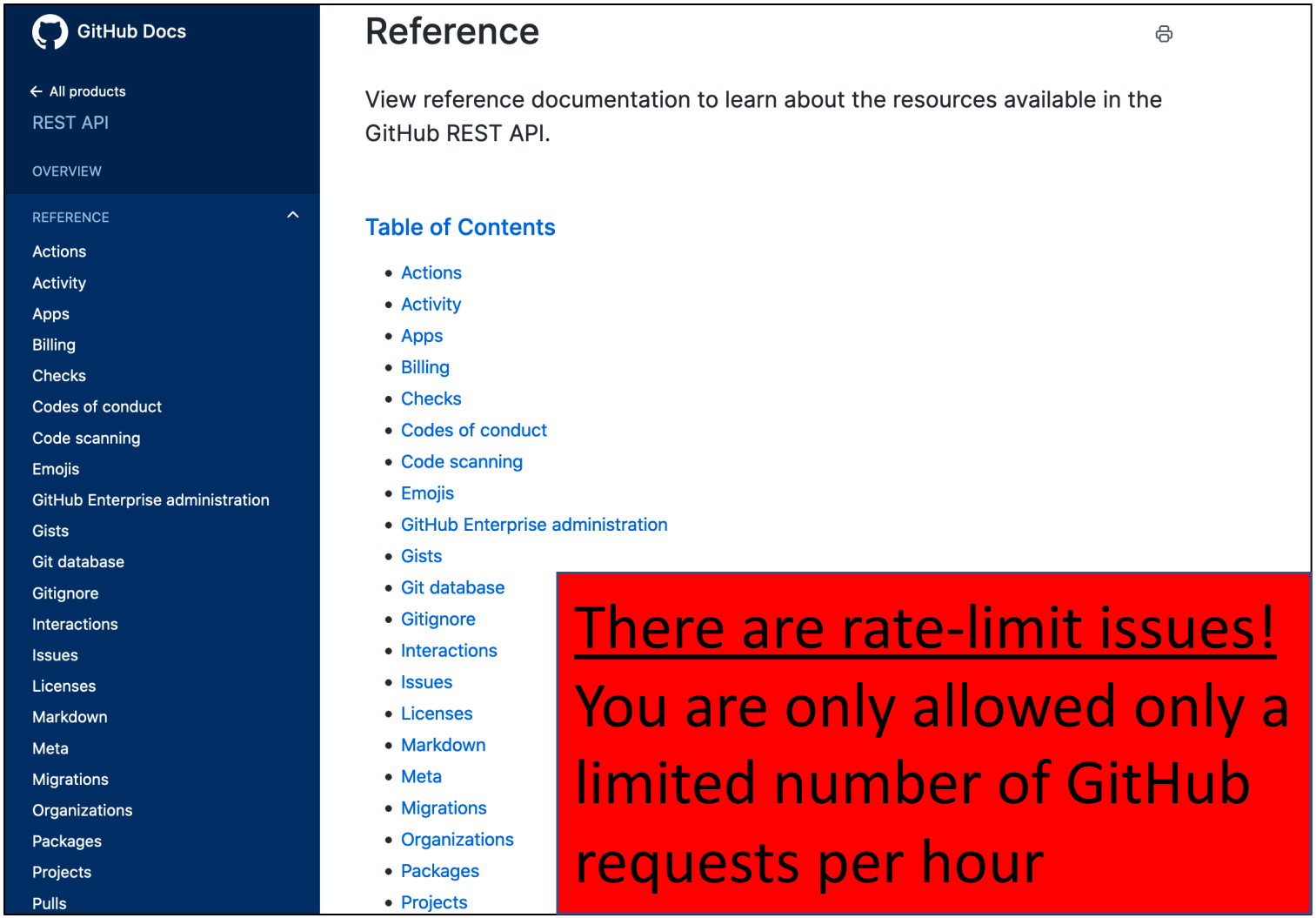
Name	Type	In	Description
`accept`	string	header	Setting to `application/vnd.github.v3+json` is recommended.
`owner`	string	path	
`repo`	string	path	
`anon`	string	query	Set to `1` or `true` to include anonymous contributors in results.
`per_page`	integer	query	Results per page (max 100).
`page`	integer	query	Page number of the results to fetch.

- List organization repositories
- Create an organization repository
- Get a repository
- Update a repository
- Delete a repository
- Enable automated security fixes
- Disable automated security fixes
- List repository contributors
- Create a repository dispatch event
- List repository languages
- List repository tags
- List repository teams
- Get all repository topics
- Replace all repository topics
- Transfer a repository
- Check if vulnerability alerts are enabled for a repository
- Enable vulnerability alerts
- Disable vulnerability alerts
- Create a repository using a template
- List public repositories
- List repositories for the authenticated user
- Create a repository for the authenticated user
- List repositories for a user

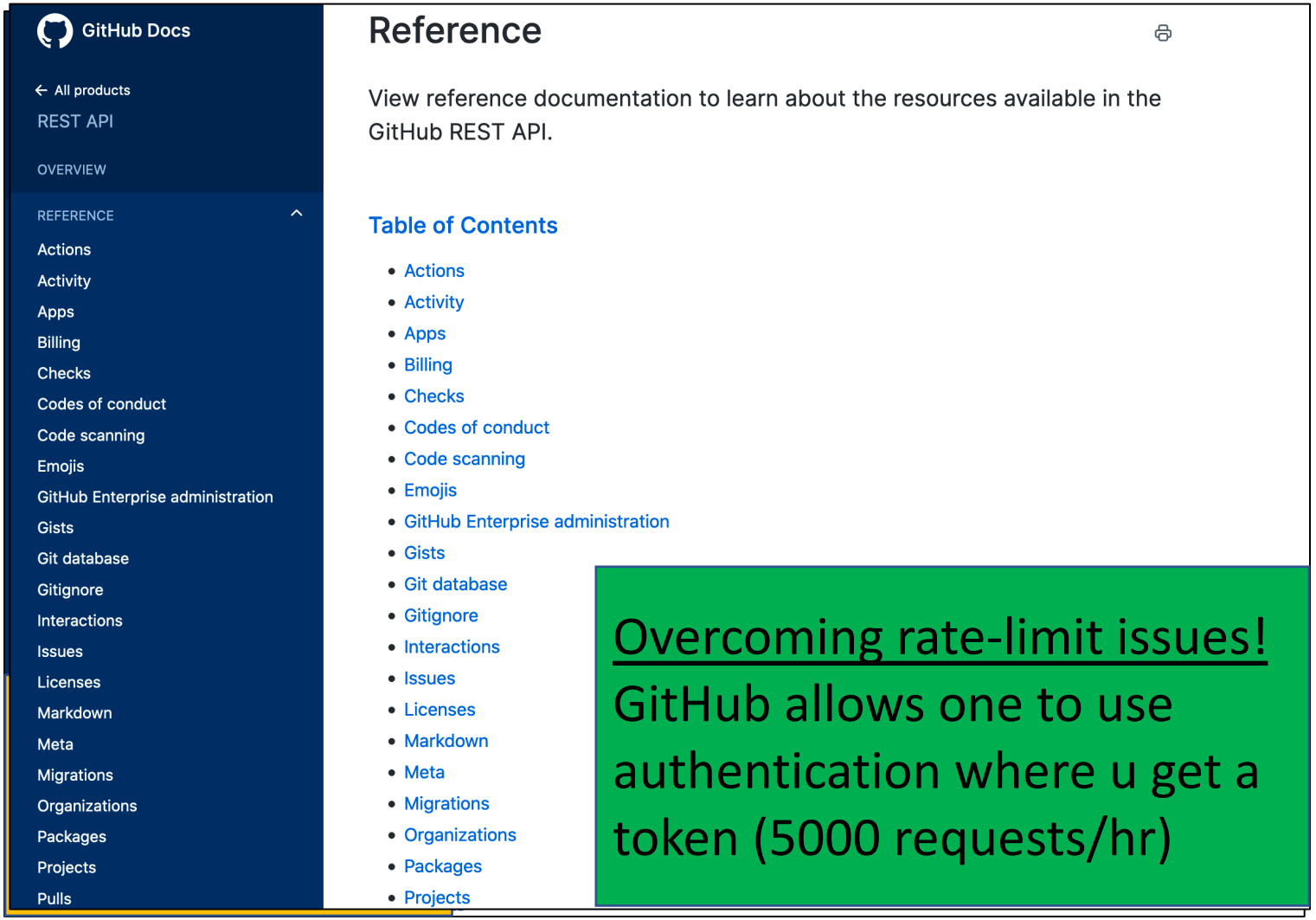
Branches

- List branches
- Get a branch
- Get branch protection
- Update branch protection

How can we mine GitHub Data



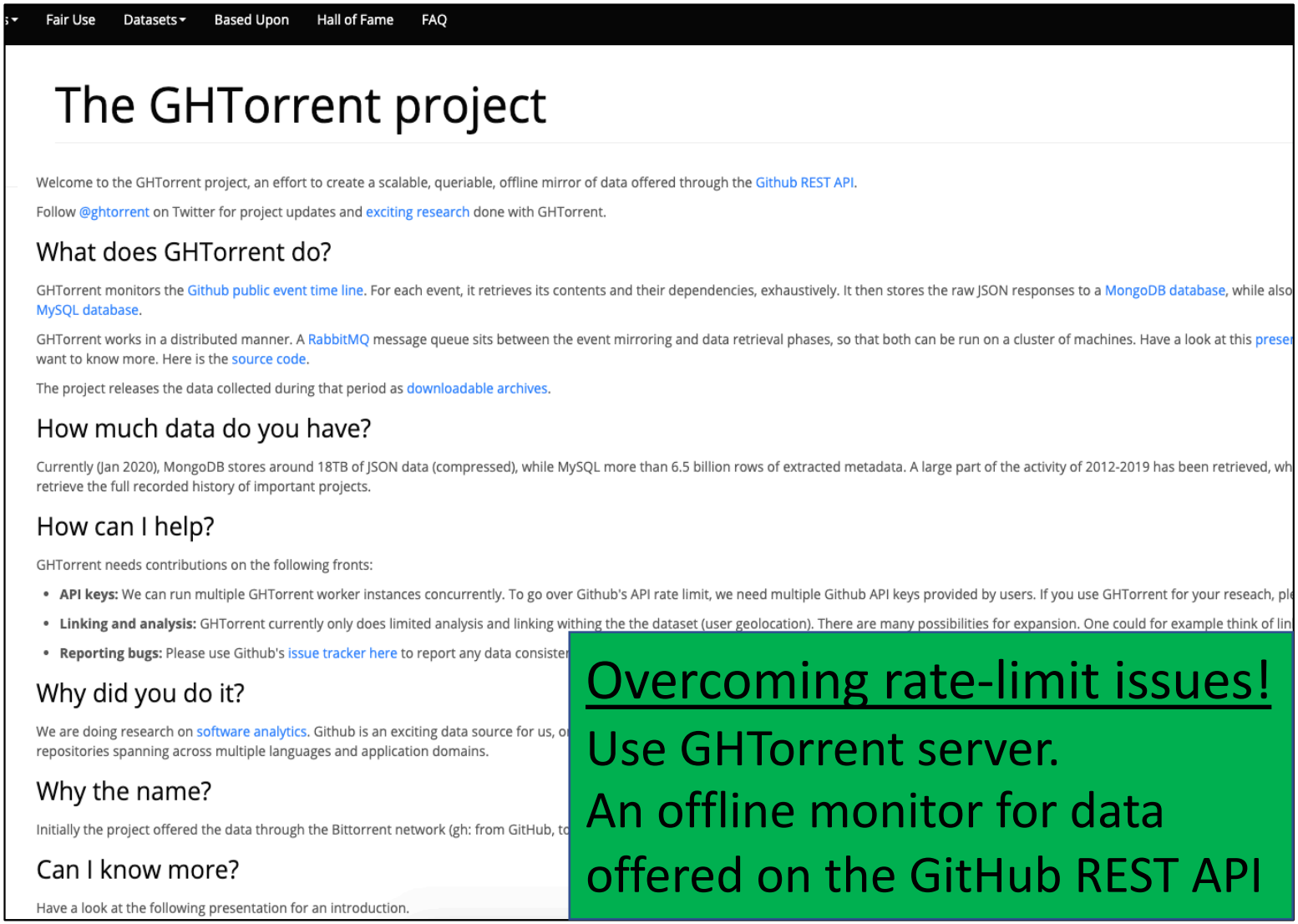
How can we mine GitHub Data



Overcoming rate-limit issues!

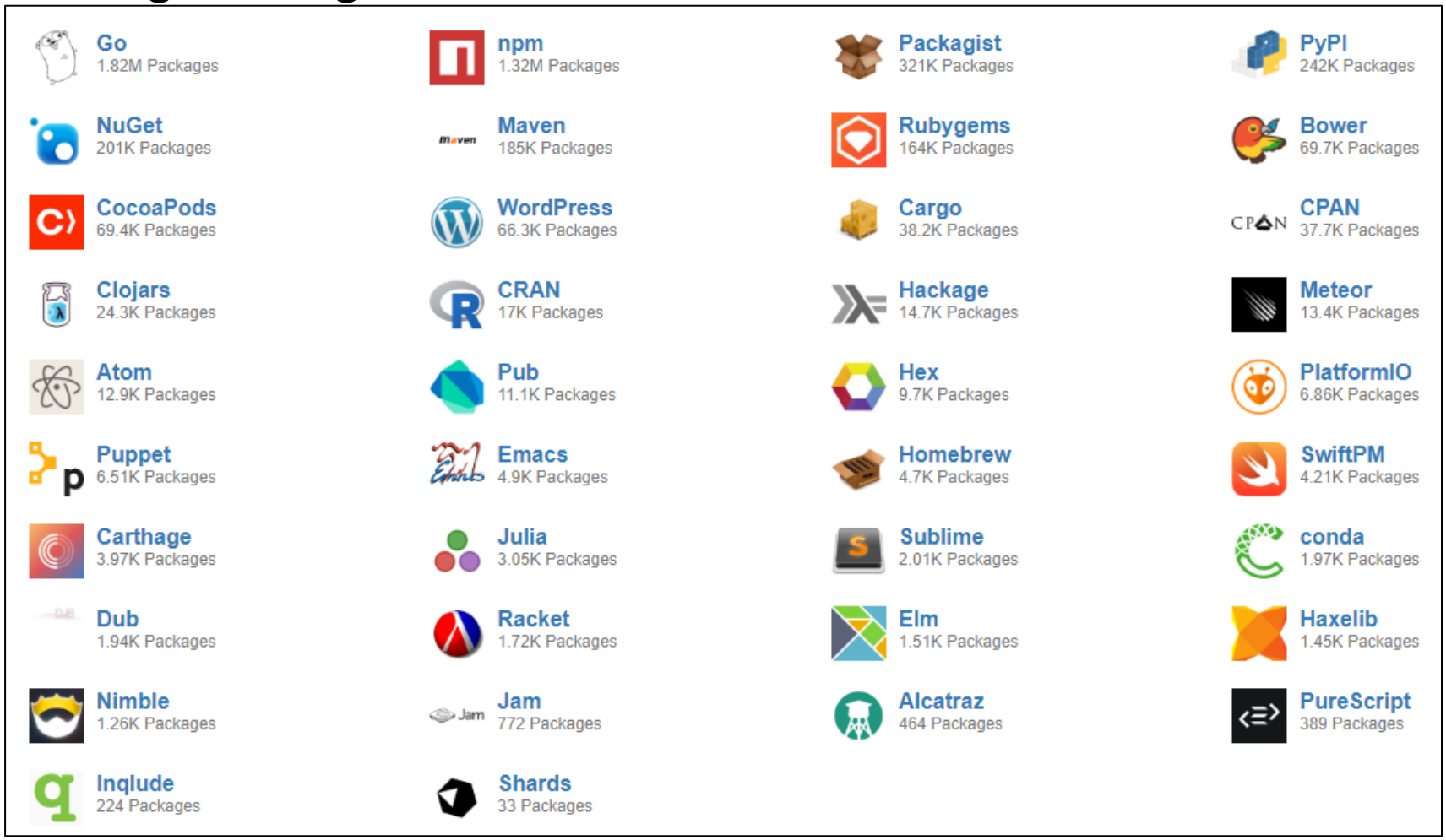
GitHub allows one to use authentication where u get a token (5000 requests/hr)

How can we mine GitHub Data

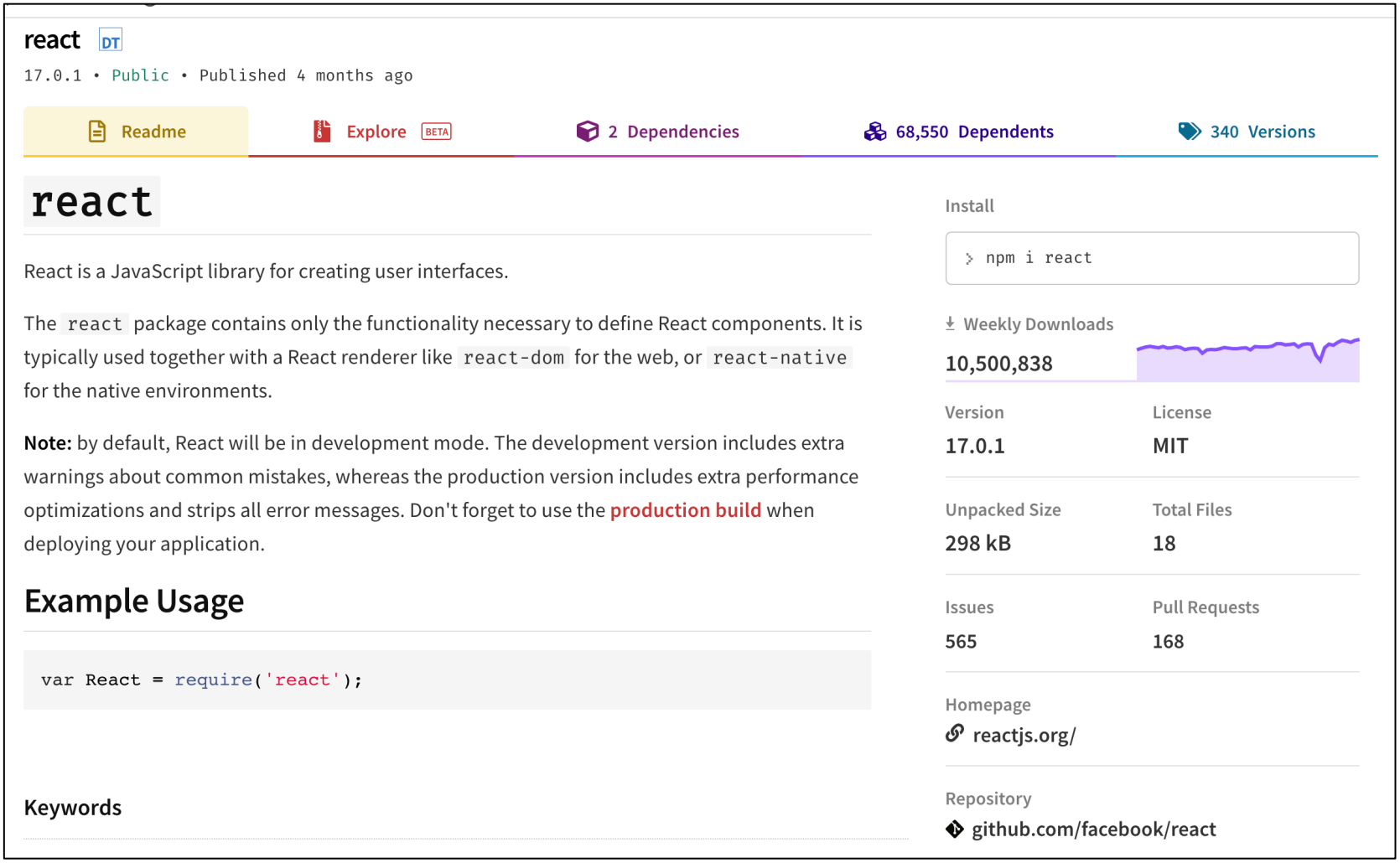


How can we mine SE Data – Part II

Package managers - Libraries.io



How can we mine SE Data from package managers?



react DT

17.0.1 • Public • Published 4 months ago

Readme | Explore BETA | 2 Dependencies | 68,550 Dependents | 340 Versions

react

React is a JavaScript library for creating user interfaces.

The `react` package contains only the functionality necessary to define React components. It is typically used together with a React renderer like `react-dom` for the web, or `react-native` for the native environments.

Note: by default, React will be in development mode. The development version includes extra warnings about common mistakes, whereas the production version includes extra performance optimizations and strips all error messages. Don't forget to use the production build when deploying your application.

Example Usage

```
var React = require('react');
```

Keywords

Install

```
> npm i react
```

Weekly Downloads

10,500,838

Version	License
17.0.1	MIT

Unpacked Size	Total Files
298 kB	18

Issues	Pull Requests
565	168

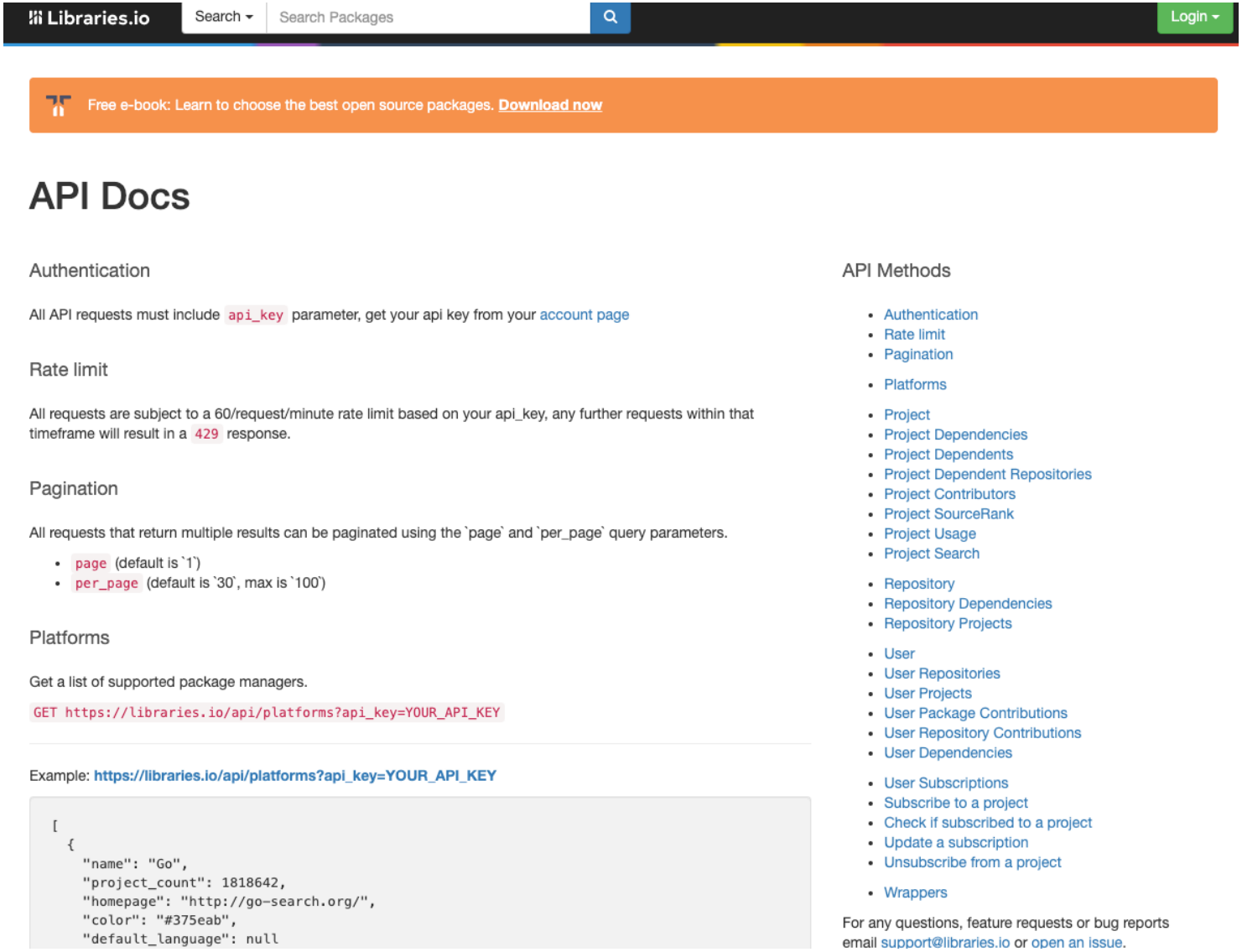
Homepage

reactjs.org/

Repository

github.com/facebook/react

How can we mine SE Data from package managers?



Let’s get our hands dirty with the labs exercise 😉