

Hotspotter

1 Introduction

The Hotspotter web application is used to clone public git-based repositories, extract metadata from the repository such as the file structure, time of and number of commits, and the authors of commits on every file within the repository. This data is then crawled through by our scoring algorithm. This algorithm produces a score for each file at each time division (10 divisions by default). The score takes into consideration the time and frequency of commits recorded for a particular file. The more recently a file has undergone changes, the higher its score will be. When modifications to the file cease, the file's score will "cooldown" in the following divisions. Previous research^{1,2} indicates that such a score can be used to accurately predict bugs (or "hotspots") within a project.

2 Dashboard

The dashboard is the first view encountered when using the Hotspotter application.

The screenshot shows the Hotspotter dashboard interface. At the top, there is a section labeled "Repository:" with a text input box containing the URL "https://github.com/example/repository.git". Below the input box is a blue button labeled "Add Repository". Underneath this is a section labeled "Database:". It contains a list of three repository entries, each with a "Select" button to its right. The first entry is "https://github.com/axel-zarate/angular-tree-view.git" with sub-statuses "Clone: True", "Score: In progress", and "Scan: In progress". The second entry is "https://github.com/eu81273/angular.treeview.git" with sub-statuses "Clone: True", "Score: True", and "Scan: True". The third entry is "https://github.com/caolan/async.git" with sub-statuses "Clone: True", "Score: False", and "Scan: True".

Repository	Clone	Score	Scan
https://github.com/axel-zarate/angular-tree-view.git	True	In progress	In progress
https://github.com/eu81273/angular.treeview.git	True	True	True
https://github.com/caolan/async.git	True	False	True

If any repositories have been submitted, they will be listed under the "Database" heading. A new repository can be added by pasting the full URL in the "Repository" input box. This field only accepts complete HTTP URLs to files that end with a ".git" extension. Pressing the "Add Repository" button will initiate the cloning and scanning processes on the server.

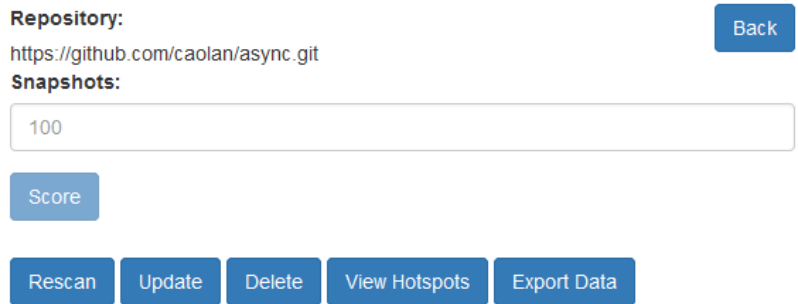
For each repository listed, you can see if it has been Cloned, Scored, and/or Scanned successfully. Each property moves from False, to In Progress, to True. Cloning denotes whether the repository has been successfully cloned to the server. Score denotes if the scoring algorithm has been completed on the repository. Scan denotes if the repository's metadata has been extracted. Cloning and Scanning occur automatically upon submission of the repository. Scoring must be initiated from the repository's options page.

¹ Lewis, Chris, and Rong Ou. "Bug Prediction at Google." *Bug Prediction at Google*. 14 Dec. 2011.

² Rahman, Foyzur, et al. "BugCache for inspections: hit or miss?." *Proceedings of the 19th ACM SIGSOFT symposium and the 13th European conference on Foundations of software engineering*. ACM, 2011.

3 Repository Options

By clicking the “Select” button next to a repository, we see the options available. These options are available for any repository. To travel back to the previous view, a “Back” button is provided in the top right.



Repository: <https://github.com/caolan/async.git> Back

Snapshots: Score

Rescan Update Delete View Hotspots Export Data

The “Snapshots” input refers to the time divisions explained in the introduction of this manual. A

higher number of snapshots gives a finer-grained view of each file’s scores over time. However, the time necessary to score the repository increases directly with the number of snapshots. After inputting an integer number of snapshots, pressing the “Score” button will begin the scoring process.

The remaining options are explained below.

- **Rescan**
 - This redoes the scanning process of extracting metadata from the repository. This is only necessary if the initial, automatic scan fails for an unexpected reason.
- **Update**
 - Pressing Update causes the server to perform a pull request on the selected repository, ensuring that the version scanned and scored is the latest. This is equivalent to deleting and re-submitting the repository.
- **Delete**
 - Deleting a repository removes all local files from the server and erases all scoring and metadata saved in the database.
- **View Hotspots**
 - Viewing Hotspots produces a visual, color-coded representation of the scored file structure. Files with a higher score appear redder (hotter) than their cooler counterparts. Graphs of each file’s score over time (according to the given number of snapshots, or a default of 10) can also be viewed.
- **Export Data**
 - A new window will be opened displaying the plaintext comma-separated values of the repository data. Each line represents a file in the repository. The columns contain the filepath and filename, total number of commits recorded for the file, a timestamp of the file’s most recent modification, and a score of the file at each snapshot. This page should be saved as a .CSV file for export.

4 Viewing Hotspots

The “View Hotspots” button from the previous section brings you to the following visual representation of the scored repository.

The file structure can be explored by clicking on folders to expand their contents. Each file’s color indicates its score where the higher score (normalized from 0 to 1) is hotter than lower scores. The arrows to the left and right of the “Snapshot” allow you to specify the cut-off point for the scores displayed. For example, by traveling back to the fifth snapshot, the scores displayed are those that the repository would have halfway through its lifespan. This allows you to quickly see how the files’ scores progress over time.

By selecting a file, the exact score can be seen as well as some brief details about the file.

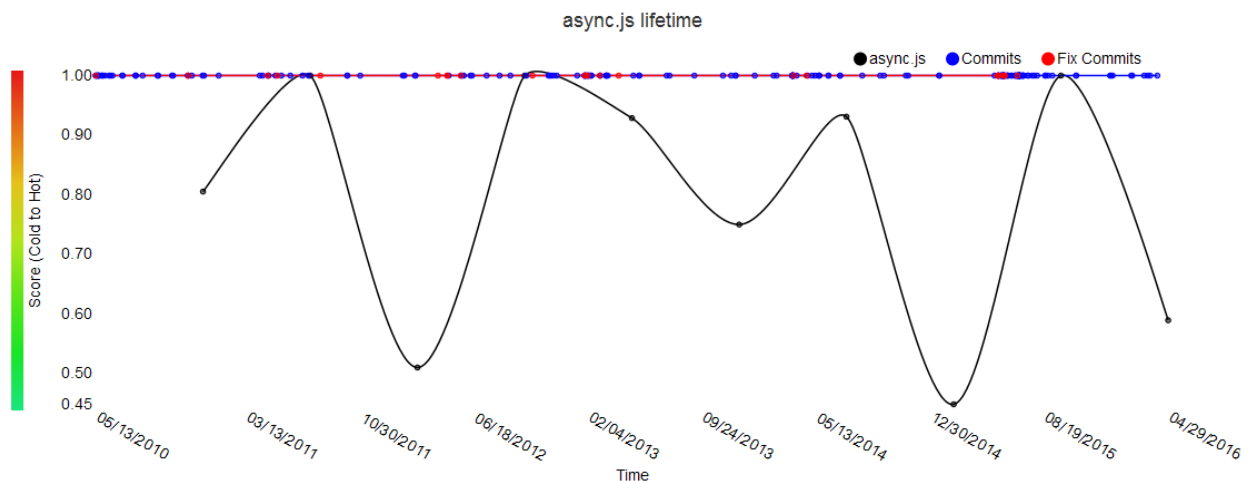
async.js

58.9% Likelihood of Containing Bugs
Last Touched By: Alexander Early
Thu Apr 07, 2016 - 4:10PM

Graph



A graph of the file’s scores over time for all of the snapshots can be seen by pressing the Graph button.



The black line indicates the score over time. At the top of the graph, data points are placed indicating when commits are made. A red data point indicates a “Fix Commit”, a commit whose message body contains keywords used by various issue-tracking software to determine if the commit was intended to resolve an existing bug.