
RepoSherlock

Release 0.1.0

Omar Elazhary

Jan 19, 2021

CONTENTS

1	RepoSherlock	1
1.1	Introduction	1
1.2	How Do I Use RepoSherlock?	1
1.3	How Do I Install RepoSherlock?	2
2	Indices and tables	3

REPOSHERLOCK

1.1 Introduction

RepoSherlock is a package intended to facilitate retrieving information from software repositories hosted by services such as GitHub, GitLab, BitBucket, or similar services. It uses their APIs to fetch issue, pull/merge request, and commit data for further processing.

1.1.1 Who is this for?

I developed this primarily for researchers interested in studying software repositories, since it helps me with my work too. While it was originally meant for research purposes, it is conceivable that it would have other applications as well. As such, this project is licensed under the MIT license.

1.1.2 What services does RepoSherlock support queries for?

So far, RepoSherlock supports data extraction from repositories hosted on GitHub. I have plans to port my previous BitBucket client to this project too, and eventually, add a GitLab client as well. Stay up-to-date by monitoring RepoSherlock's [issues page](#).

1.2 How Do I Use RepoSherlock?

RepoSherlock supports two modes of use: - as a standalone application, and - as a module within your python script.

1.2.1 As a standalone application

Once installed, you can use RepoSherlock as a standalone application in your terminal of choice. It supports the following arguments:

- **-h, -help**: Shows a help message and exits.
- **-user <username>**: Your username on the service from which you want to retrieve data.
- **-token <token>**: The token provided to you by your repository service.
- **-target <owner/repository>**: The repository whose data you want to pull.
- **-type [GitHub|BitBucket]**: Your repository management service. Currently, only GitHub is supported.
- **-outdir <path/to/output/directory>**: The output directory where RepoSherlock should save the queried data.

- **--pages <number_of_pages>**: The maximum number of pages of data to fetch. Default is 1000. Naturally, RepoSherlock will stop once no more data is available.

A typical command using RepoSherlock looks like the following example:

```
$ reposherlock --user omazhary \  
--token <long_alphanumeric_token> \  
--target omazhary/reposherlock \  
--type GitHub \  
--outdir output/.
```

1.2.2 As a Python module

You can build a python script and use RepoSherlock within it to fetch data on the fly to do with as you please. For instance, if you wanted to create a GitHub client to use in your python script, you would import it as a dependency, and give it the necessary information:

```
from reposherlock.github import GitHub  
  
client = GitHub('omazhary', 'my_long_alphanumeric_token')  
issues = client.get_issues('omazhary/reposherlock', 1000)  
pull_requests = client.get_pull_requests('omazhary/reposherlock', 1000)  
commits = client.get_commits('omazhary/reposherlock', 1000)
```

Further documentation can be found [here](#).

1.3 How Do I Install RepoSherlock?

Regardless of whether you want to use it as a standalone application or a module in your project, you can install RepoSherlock via pip as you would any normal python module:

```
$ pip install reposherlock
```

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`