

How do I use Sourcegraph with Ruby?

Satish Talim

This book is for sale at <http://leanpub.com/howdoiusesourcegraphwithruby>

This version was published on 2014-06-16



This is a [Leanpub](#) book. Leanpub empowers authors and publishers with the Lean Publishing process. [Lean Publishing](#) is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.



This work is licensed under a [Creative Commons Attribution 3.0 Unported License](#)

Tweet This Book!

Please help Satish Talim by spreading the word about this book on [Twitter!](#)

The suggested hashtag for this book is [#sourcegraphruby](#).

Find out what other people are saying about the book by clicking on this link to search for this hashtag on Twitter:

<https://twitter.com/search?q=#sourcegraphruby>

Also By **Satish Talim**

[How Do I Write And Deploy Simple Web Apps With Go?](#)

[Building a package in Go](#)

[How do I use Sourcegraph with Go?](#)

Contents

Preface	i
Who is the booklet for?	i
Acknowledgements	i
Using Code Examples	i
How to Contact Me	i
Thanks	i
1 What's Sourcegraph?	1
1.1 Getting Started	1
1.2 How Do I Use It?	1
1.3 Assumptions	1
1.4 Get Started	1
1.5 sourcegraph.rb - Outline 1	1
1.6 sourcegraph.rb - Outline 2	5
1.7 sourcegraph.rb - Outline 3	7
1.8 sourcegraph.rb - Outline 4	9
1.9 sourcegraph.rb - Final program	10

Preface

Who is the booklet for?

This short booklet will show Ruby newbies how one can use Sourcegraph to better write one's Ruby programs. To try out the programs in this booklet, you should have a working copy of Ruby 2 on your computer.

Acknowledgements

I would like to thank [Sourcegraph](https://sourcegraph.com/)¹ for permitting me to write this booklet.

Using Code Examples

All of the code in this booklet can be used pretty much anywhere and anyhow you please.

How to Contact Me

I can be reached via e-mail at satish.talim@gmail.com. Please contact me if you have any questions, comments, kudos or criticism on the booklet. Constructive criticism is definitely appreciated; I want this booklet to get better through your feedback.

Thanks

Thanks for downloading and checking out this booklet. As part of the lean publishing philosophy, you'll be able to interact with me as the booklet is completed. I'll be able to change things, reorganize parts, and generally make a better booklet. I hope you enjoy.

The Ruby Logo is Copyright (c) 2006, Yukihiro Matsumoto. It is licensed under the terms of the [Creative Commons Attribution-ShareAlike 2.5 agreement](http://creativecommons.org/licenses/by-sa/2.5/)².

¹<https://sourcegraph.com/>

²<http://creativecommons.org/licenses/by-sa/2.5/>

1 What's Sourcegraph?

Sourcegraph is a code search engine that shows you documentation and real-world usage examples for hundreds of thousands of libraries written in Go and Ruby.

1.1 Getting Started

[Sign up](#)¹ with your GitHub account (no private data is requested). Signing in is optional, but it helps Sourcegraph find all of your open-source code and attribute it to you.

1.2 How Do I Use It?

We shall build a small Ruby program and use Sourcegraph along the way. This simple application: given a subreddit like `ruby`, it fetches that subreddit's author of an article and the url of that article.

1.3 Assumptions

I am assuming that you have downloaded and installed Ruby.

1.4 Get Started

Open a command window, make a new folder and cd to it as follows:

```
$ mkdir sourcegraph
$ cd sourcegraph
```

1.5 sourcegraph.rb - Outline 1

I have a very basic outline of the code `sourcegraph.rb`.

Program `sourcegraph.rb`

```
if ARGV.length != 1
  abort("Usage: ruby sourcegraph.rb ruby")
end
```

We shall be accepting the subreddit name as a command-line argument to our program. If we want to see the subreddit `ruby`, we shall run our program by typing:

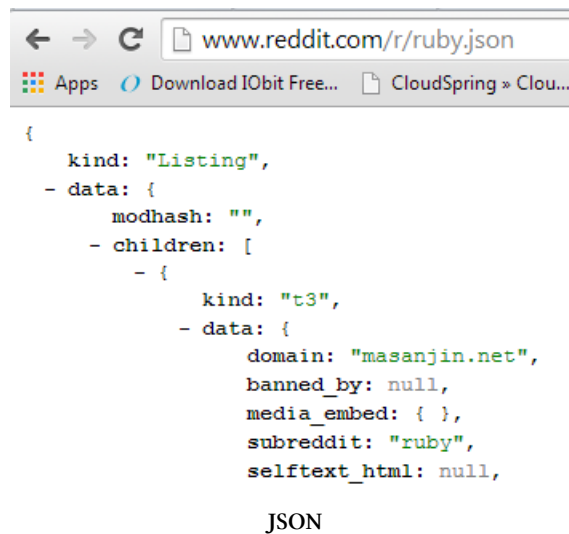
¹<https://sourcegraph.com/join>

```
ruby sourcegraph.rb ruby
```

`ARGV` contains the arguments passed to our script, one per element.

I want to make a `Http` request to the Reddit API. Next I would like to parse the JSON response and determine the subreddit's author of an article and the url of that article.

In your browser, open the site <http://reddit.com/r/ruby.json> the browser output is a huge blob of JSON that we receive from the Ruby Subreddit. This may be difficult to look at in the browser, unless you have the JSONView plugin installed. These extensions are available for [Firefox](#)² and [Chrome](#)³. With the extension installed, here's a partial view of the JSON:



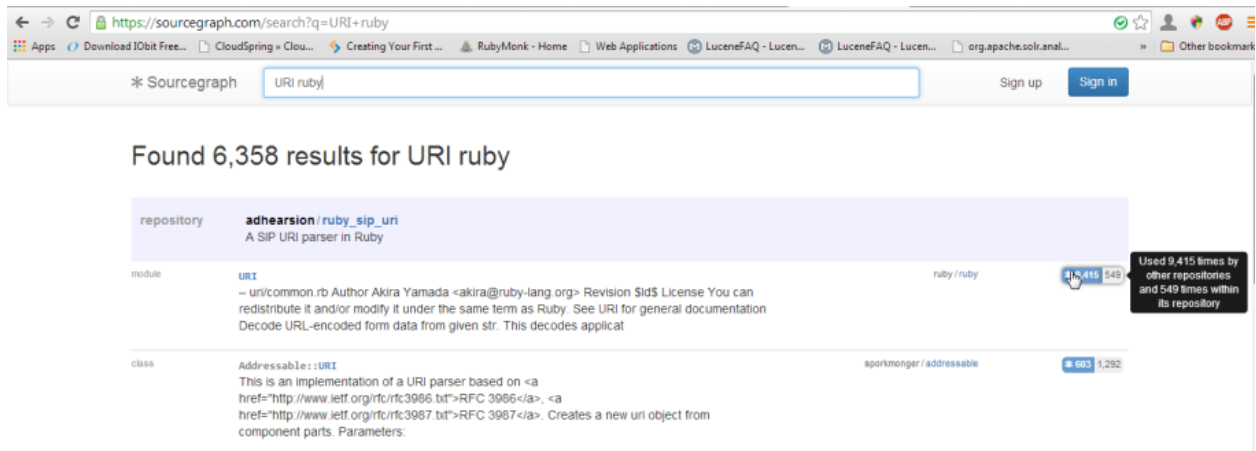
In my program, I want to convert the string `http://reddit.com/r/ruby.json` to a properly formatted Uniform Resource Identifier using the `URI` module. However, I haven't used this module much and bad at remembering how to exactly use it. I would definitely like to know how. So, let us look it up on [sourcegraph](#)⁴.

²<https://addons.mozilla.org/en-us/firefox/addon/jsonview/>

³<https://chrome.google.com/webstore/detail/jsonview/chklaanhfefbnpoihckbnefhakgolnmc>

⁴<https://sourcegraph.com/>

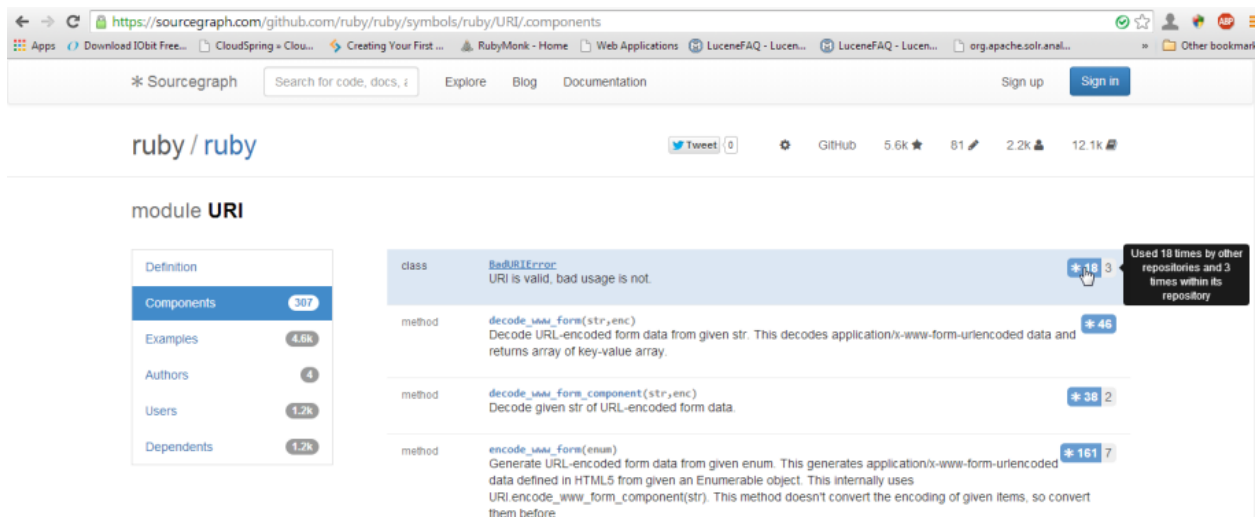
Let's type **URI** **ruby** as shown below:



Sourcegraph

If you see **module URI** there are over 9,000 examples of usage of **URI** on sourcegraph. That's good. Let's click on that. Next click on the "Components" link on the left.

In the image below, you can see the functions and other definitions in the module **URI**. On the right you can see how many times it has been used by other people.



Sourcegraph

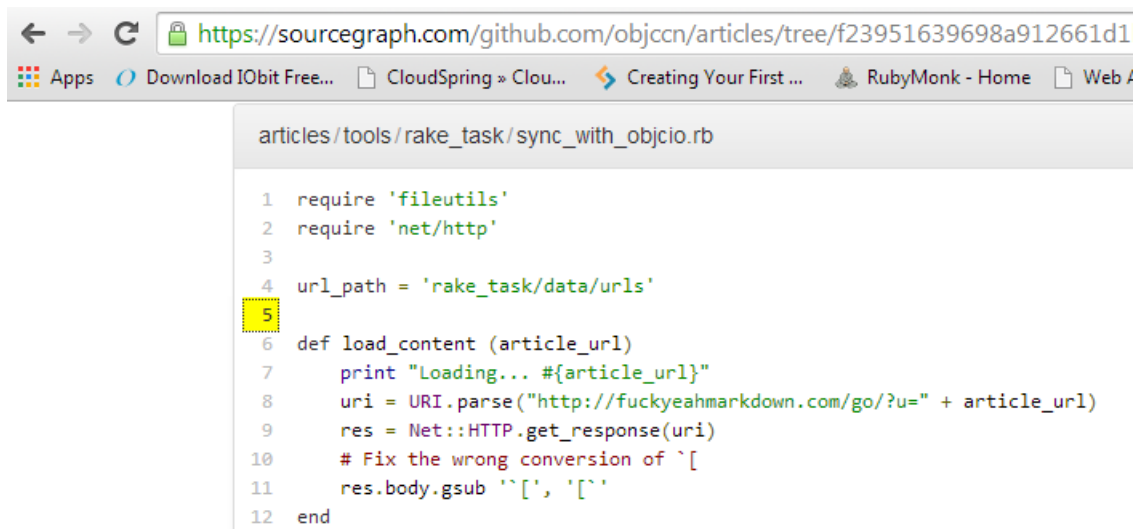
Click on the Examples link on the left and see some quick examples of how it is used.

Scroll down and see which example is similar to the one you want to write. I think **sync_with_objcio.rb** seems to be what I want.



Sourcegraph

Click on `sync_with_objcio.rb` to load the full example as seen in the image below.



Sourcegraph

I think I will use `URI.parse` in my program `sourcegraph.rb`.

1.6 sourcegraph.rb - Outline 2

Let's type in the program as follows:

Program `sourcegraph.rb`

```
require 'net/http'

if ARGV.length != 1
  abort("Usage: ruby sourcegraph.rb ruby")
end

url = "http://www.reddit.com/r/#{ARGV[0]}.json"
uri = URI.parse(url)
puts uri
```

Note that in the program above, we are using the `net/http` package.

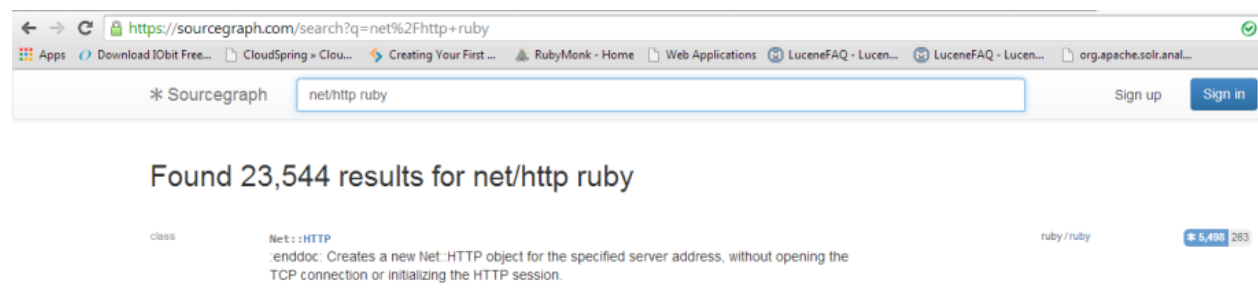
Next, in the same folder where the program is located, type:

```
$ ruby sourcegraph.rb ruby
```

We get a properly formatted URI:

```
http://www.reddit.com/r/ruby.json
```

Cool! It's working. Now, let us write the code that fetches from the Reddit API. Let's go back to Sourcegraph and see how we can do this.



Sourcegraph

Let's click on `net/http`. We should see as shown in the image below:

The screenshot shows the Sourcegraph interface for the `Net::HTTP` class. The left sidebar contains a table of contents with the following items:

Item	Count
Definition	
Components	131
Examples	3k
Authors	13
Users	807
Dependents	809

The main content area displays the documentation for `Net::HTTP`. It includes a summary: "Net::HTTP is a Ruby class. It is created by 13 authors. It consists of 131 components. It is used 3,015 times by 807 programmers in 809 projects." Below this is the "Documentation" section, which contains the following text:

`:enddoc:`

Creates a new Net::HTTP object for the specified server address, without opening the TCP connection or initializing the HTTP session. The `address` should be a DNS hostname or IP address.

The default port to use for HTTP requests; defaults to 80.

Sends a GET request to the target and returns the HTTP response as a string. The target can either be specified as (`uri`), or as (`host`, `path`, `port` = 80); so:

```
print Net::HTTP.get(URI('http://www.example.com/index.html'))
```

Sourcegraph

In the image, we can see the documentation, the `net/http` being used by 807 programmers and so on. Let's click on Examples on the left.

The screenshot shows the Sourcegraph interface for the examples of the `Net::HTTP` class. The left sidebar is the same as in the previous image, but the "Examples" item is highlighted. The main content area displays the code from the file `objccn/articles/articles/tools/rake_task/sync_with_objcio.rb`. The following code is highlighted:

```
uri = URI.parse("http://fuckyeahmarkdown.com/go/?u=" + article_url)
res = Net::HTTP.get_response(uri)
# Fix the wrong conversion of `[`
res.body.gsub ']', '['
```

Sourcegraph

Oh! Observe that the same example `sync_with_objcio.rb` contains the code that we need. I have highlighted the required code which I will copy/paste into my program.

1.7 sourcegraph.rb - Outline 3

Program sourcegraph.rb

```
require 'net/http'

if ARGV.length != 1
  abort("Usage: ruby sourcegraph.rb ruby")
end

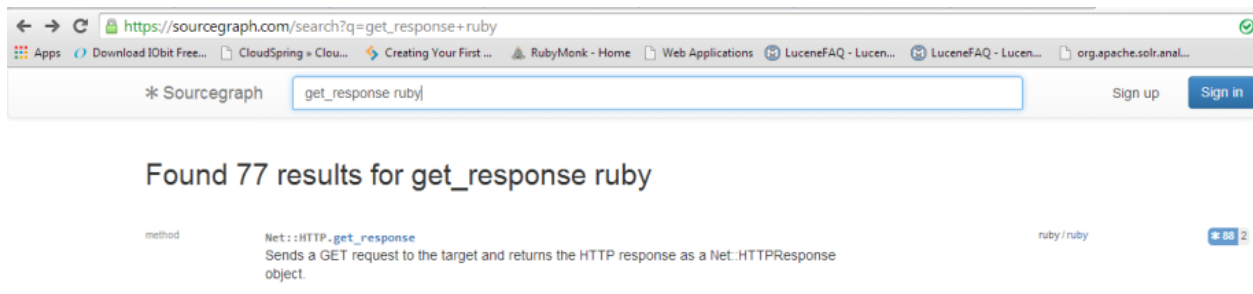
url = "http://www.reddit.com/r/#{ARGV[0]}.json"
uri = URI.parse(url)

res = Net::HTTP.get_response(uri)
```

In this program we are able to get the response **res** but how do we get it's contents?

We had copied/pasted the code from the **sync_with_objcio.rb** program. Let's go back to its full program listing and check if we find what to do with **res**. You will soon realize that there's nothing related to **res** there. Let us go back to the other examples listed where **sync_with_objcio.rb** is. Scroll down. Oh! Nothing here.

Let's search again on Sourcegraph for **get_response ruby**.



Sourcegraph

Click on **get_response**.

The screenshot shows the Sourcegraph web interface for the `Net::HTTP.get_response` method. The browser address bar shows the URL `https://sourcegraph.com/github.com/ruby/ruby/symbols/ruby/Net/HTTP/$classmethods/get_response`. The page header includes the Sourcegraph logo, a search bar, and navigation links like 'Explore', 'Blog', and 'Documentation'. Below the header, the page title is 'ruby / ruby'. The main content area displays the method signature `method Net::HTTP.get_response(uri_or_host,path,port,&block)`. On the left, there is a sidebar with links to 'Definition', 'Examples' (87), 'Authors' (4), 'Users' (66), and 'Dependents' (64). The main content area shows the definition of the method, stating it is a Ruby method created by 4 authors, used 87 times by 66 programmers in 64 projects. It includes a 'Documentation' section with a description: 'Sends a GET request to the target and returns the HTTP response as a Net::HTTPResponse object. The target can either be specified as (uri), or as (host, path, port = 80); so:'. Below the description is a code snippet: `res = Net::HTTP.get_response(URI('http://www.example.com/index.html'))` and `print res.body`.

Sourcegraph

I observe that I can extract the `res.body` out of `res`.

Let's add `print res.body` to our program and run it.

Program sourcegraph.rb

```
require 'net/http'

if ARGV.length != 1
  abort("Usage: ruby sourcegraph.rb ruby")
end

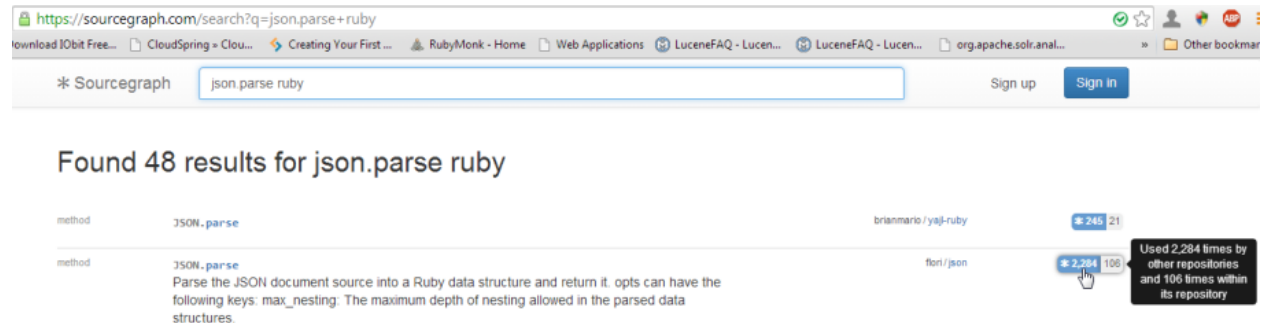
url = "http://www.reddit.com/r/#{ARGV[0]}.json"
uri = URI.parse(url)

res = Net::HTTP.get_response(uri)
print res.body
```

I see a huge blob of JSON in the `res` object. I need to parse this JSON and get what I want. How?

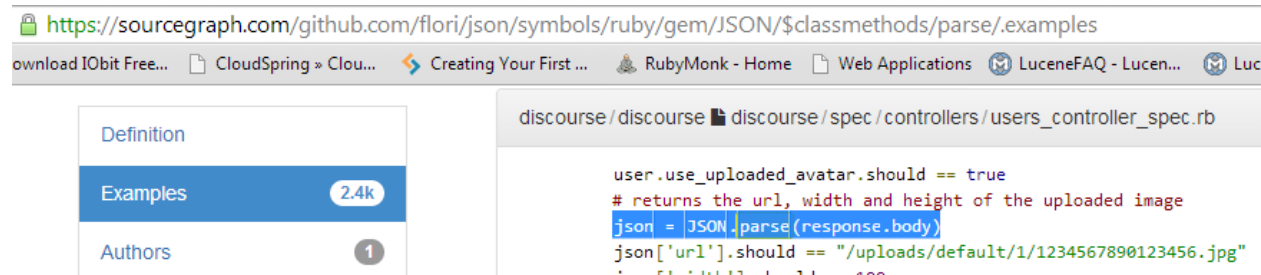
1.8 sourcegraph.rb - Outline 4

Going back to Sourcegraph, let me search for `json.parse ruby`.



Sourcegraph

Click on the method `parse` and then on the screen that comes up, click on “Examples”. Scroll down and see which example is similar to the one you want to write. I think `users_controller_spec.rb` seems to be what I want.



Sourcegraph

The highlighted code is returning us a JSON object.

Let us copy/paste the relevant code from `users_controller_spec.rb` into our program:

Program sourcegraph.rbo

```
require 'json'
require 'net/http'

if ARGV.length != 1
  abort("Usage: ruby sourcegraph.rb ruby")
end

url = "http://www.reddit.com/r/#{ARGV[0]}.json"
uri = URI.parse(url)

res = Net::HTTP.get_response(uri)
result = JSON.parse(res.body)
print result
```

Observe that we need the statement `require 'json'`.

Run the program and you can see the API response, but we want to show some specific pieces of information. We can see the `author` and `url` in that data.

1.9 sourcegraph.rb - Final program

In your browser, again open the site `http://reddit.com/r/ruby.json` the browser output is formatted JSON (because of the plugin that we had installed earlier).

Observe that `result['data']['children']` is an Array of Hash that contains many `['author']` and `['url']`.

We shall use a `for` loop to extract all the `['author']` and `['url']`, as shown in the program below:

Program `sourcegraph.rb`

```
require 'json'
require 'net/http'

if ARGV.length != 1
  abort("Usage: ruby sourcegraph.rb ruby")
end

url = "http://www.reddit.com/r/#{ARGV[0]}.json"
uri = URI.parse(url)

res = Net::HTTP.get_response(uri)
result = JSON.parse(res.body)

# result['data']['children'] is an Array of Hash
for x in 0..(result['data']['children'].length-1)
  puts "Author: " + result['data']['children'][x]['data']['author']
  puts "Article URL: " + result['data']['children'][x]['data']['url']
  puts ""
end
```

Re-run the program:


```
$ ruby sourcegraph.rb ruby
```

```
Author: daviddddh
```

```
Article URL: https://github.com/DavidHuie/quartz
```

```
Author: Categoria
```

```
Article URL: http://masanjin.net/blog/fibers
```

```
Author: egisatoshi
```

```
Article URL: http://www.egison.org/blog/ruby.html
```

That's my data and it's correct. Let us try another subreddit say **golang**:

```
$ ruby sourcegraph.rb golang
```

```
Author: mattetti
```

```
Article URL: http://www.golangbootcamp.com/?book
```

```
Author: natefinch
```

```
Article URL: https://github.com/natefinch/lumberjack
```

```
Author: IndianGuru
```

```
Article URL: https://leanpub.com/howdoiusesourcegraph
```

That's it! So searching on Sourcegraph you can quickly see how other programmers are doing similar work that we are doing. We thus save a lot of time by being able to look at code and see how things are actually used, instead of having to read thro' docs. Documentation is great and Sourcegraph is a great way to find docs quickly but sometimes an example is worth thousands of lines of documentation.

So try Sourcegraph out and if you have any feedback (questions, bugs etc.) for them, post them at <https://github.com/sourcegraph/sourcegraph.com/issues/new>⁵.

Sourcegraph is working really hard to make it the best tool for open source programmers.

⁵<https://github.com/sourcegraph/sourcegraph.com/issues/new>