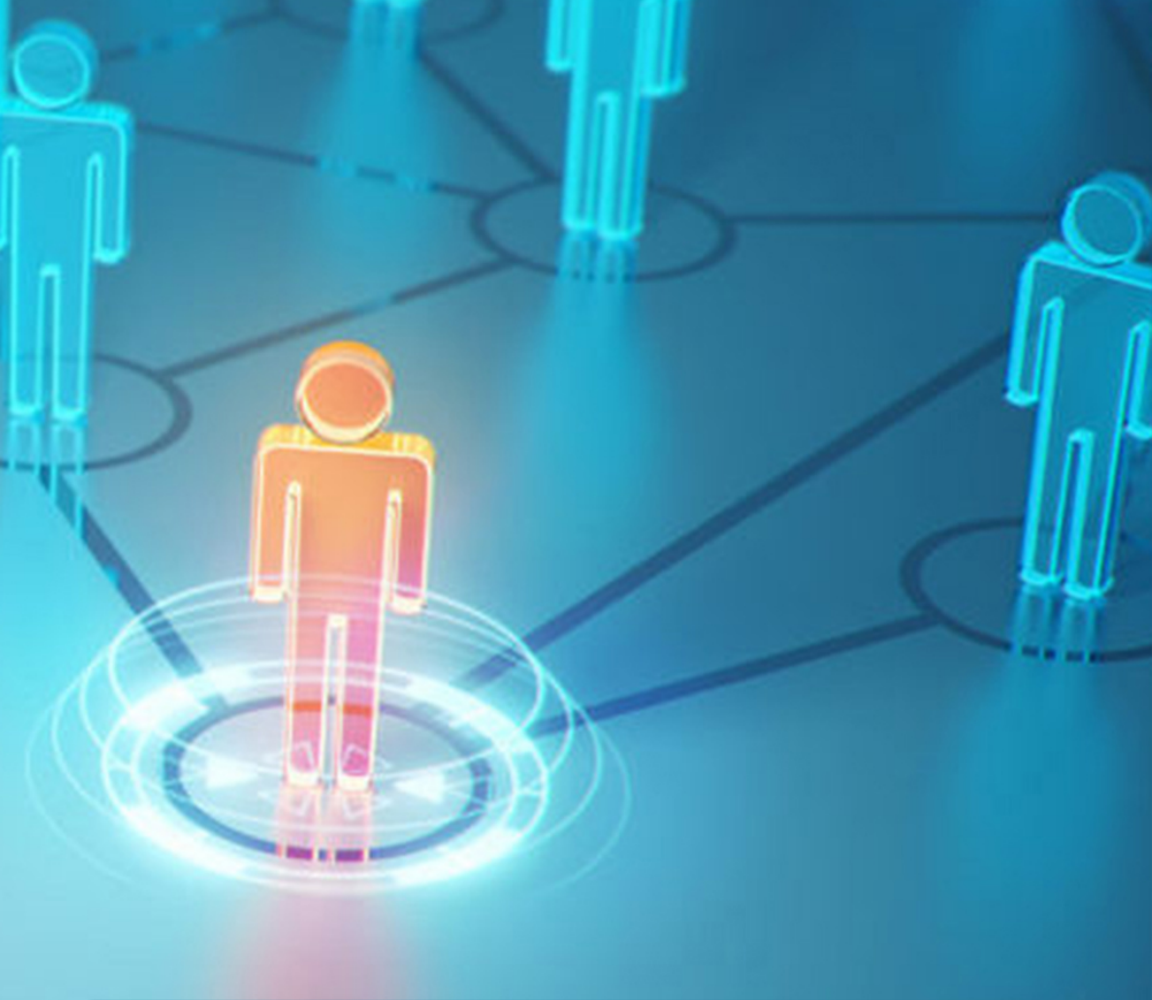


Tech Recruiting with GitHub

Find hidden candidates & win the talent search

Alex Wenz, Don Phan & Jakob Picciotto

Table of Contents

Back to top

Table of contents

Today's Flawed Recruiting Approach

Smart and resourceful recruiters rely on an array of tools and services to get the job done. Some solutions may seem like more obvious choices than others, but recruiting professionals must avoid the temptation of getting too comfortable with any one recruitment tool. In this case, we're looking at LinkedIn.

Since LinkedIn is undoubtedly today's go-to platform for most tech recruiters, don't make the all too common mistake of relying on LinkedIn alone; doing even as much as the majority of your sourcing on this one platform isn't a great idea.

We highly recommend that you act now and widen your sourcing reach to include GitHub. By all means you should keep utilizing LinkedIn, as it's undoubtedly a valuable tool.

Moreover, don't mistake our position on LinkedIn as a bash to the platform, nor advice to rid yourself of it. As the market stands today, it would be incredibly unwise to

Back to top

leave them out of your recruitment toolkit. Our goal is to simply encourage you to avoid making LinkedIn your de facto toolkit.

The professional networking giant has blossomed into a full-blown recruiting platform in which it's almost too easy for recruiters to collectively overwhelm other members with unsolicited correspondence through InMail messages and connect invites.

Because these tools are easily accessible to so many recruiters, LinkedIn has now become an overfished pond. Other sourcing options should be evaluated and explored immediately. We believe GitHub is the most promising untapped resource, and should not be overlooked any longer.

In all fairness, LinkedIn has some insightful analytics that it's able to mine from its large database of members, which recruiters can then leverage to help better target candidates.

However, what might seem like a really valuable tool for you is the same tool available to much of your recruiting competition.

For example, the Open Candidate filter option (candidates set preferences to signal to recruiters that they are open to new opportunities) is something every other LinkedIn Recruiter license holder has access to.

So, you end up competing for the same limited selection,

Back to top

bringing us back to the predicament of too many recruiters contacting too few candidates.

So what now?

According to a 2015 survey conducted by Stack Overflow, approximately 23% of developers aren't even active on LinkedIn. So while LinkedIn may be a recruiting essential, you'll definitely benefit taking regular breaks from the platform to look elsewhere.

Enter GitHub:

Since GitHub was designed from the ground up for software developers and other techies you're likely on the hunt for, it's the elephant in the room within today's tech recruiting landscape.

GitHub is nearing 40 million members who publicly share tens of millions of source code repositories that tech recruiters can explore in their quest to better assess preferred development languages, and to gain insights on personal member interests.

This ultimately leads to more precise and personalized messaging to augment the chances of successful engagement.

Back to top

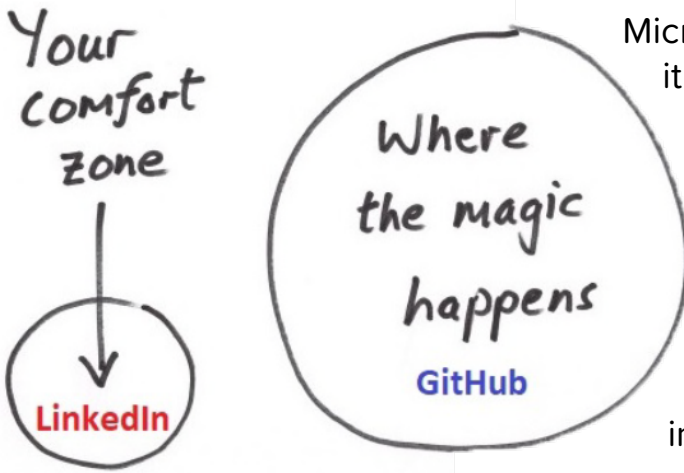
Table of contents

Why Recruit On GitHub?

If you've been in the tech talent recruitment game for any time at all, it's near impossible not to have heard of GitHub. Perhaps you've even run a few searches using the platform, only to find yourself fleeing back to the more familiar LinkedIn.

Rest assured that many other tech recruiters do the same thing, as GitHub is not designed for recruiting, and hence takes just a bit more effort to feel comfortable and productive.

In addition to being completely free to join and use by all, GitHub is by far the world's largest collaborative software developer community. Acquired by Microsoft in 2018, it is undoubtedly the go-to source code repository platform where developers create, share, improve and



Back to top

critique each other's projects.

Developers of all levels of experience and proficiency find their home on GitHub. Those sharing project source code on this popular platform range from complete novice developers to seasoned industry subject matter experts.

The key here is that there's a tremendous opportunity for tech recruiters to build a diverse pipeline of various levels of talent, meeting current and future hiring needs.

Today GitHub has well over 37 million members from hundreds of different countries (with about 11 million based in the United States alone) and well over 500 million unique visitors each month.

To compare, Stack Overflow has over 10 million registered users, as of January 2019.

With GitHub sporting these kinds of numbers, it's not hard to see why more than 50% of Fortune 50 companies have premium enterprise level GitHub accounts.

Yet, so many tech recruiters continue to overlook its value, so this is a golden opportunity in front of you to take advantage of.

Since its launch in 2008, the platform has become home to more than 100 million repositories, with at least 28 million accessible to the public.

As mentioned earlier, the Stack Overflow survey found that

Back to top

nearly 23% of developers aren't on LinkedIn.

Based on our analysis of hundreds of thousands of LinkedIn profiles, we estimate an additional 7% of users who do have accounts set up on LinkedIn have completely abandoned them.

Overall, 30% of IT workers are not on LinkedIn.

Staggering figures indeed, but it makes sense when you really think about it. More specifically, developers need to find jobs just as much as the rest of us, but once they're stable and settled in, they're focused on doing what they love: development. Techies are not generally focused on regularly selling and marketing themselves by updating their profiles on LinkedIn, DICE, Indeed, and many other employment-focused sites.

When you source technical candidates from places they *want* to be rather than where they feel like they *have* to be, you get to see them in their element.

It's candid and more down to earth. When you're able to glean what makes them tick, you waste less time sorting through what might turn out to be imperfect fits on some other platform where much less germane coding information and project interests are available for consumption. Not to mention, it'll become much easier to reach out in ways they'll respond to thanks to your newfound insight.

On top of this, when compared to LinkedIn's

Back to top

‘endorsement’ feature, peer-to-peer, coder-to-coder following speaks louder and with more validity.

To summarize, overlooking or simply under-utilizing GitHub is a very common tech recruiter mistake, but just reading this eBook is a terrific first step. Pretty soon you’ll find even minimal time and effort invested on GitHub’s platform can bring you amazing returns.

Back to top

Table of contents

Tour of GitHub

Aside from the fact that GitHub is completely free to be a part of, joining the GitHub community is absolutely key to unlocking one of the most valuable user profile data points made publicly available by many developers: their personal email. An added bonus to using the platform is that it's highly likely that a user will display their personal email address instead of or in addition to their work email address.

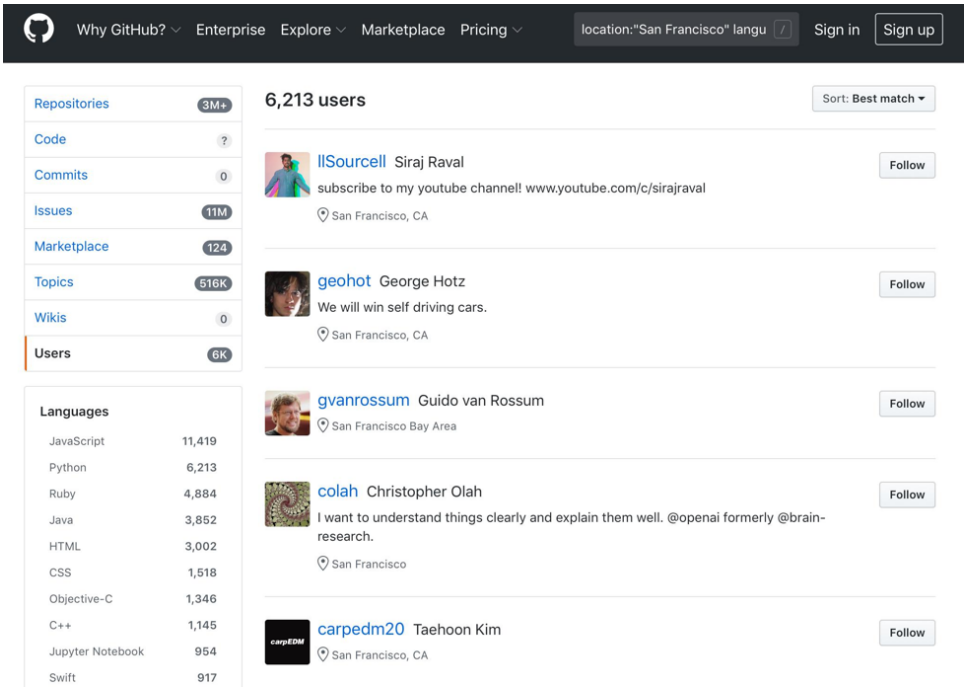
As any tech recruiter knows, personal email addresses are far more reliable when reaching out for recruitment purposes.

Since work emails tend to change over time when compared to the more static personal email addresses, personal email discovery is a major win. The ability to discover a member's email address alone is the number one no-brainer reason to create an account.

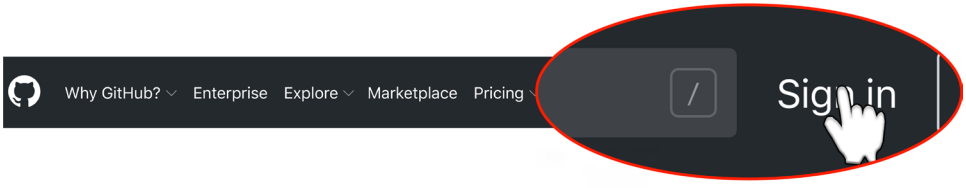
Back to top

If you choose *not* to create an account and simply run a search (more on GitHub search syntax soon), you are rewarded with plenty of member profiles.

This sample search shows 6,213 users:

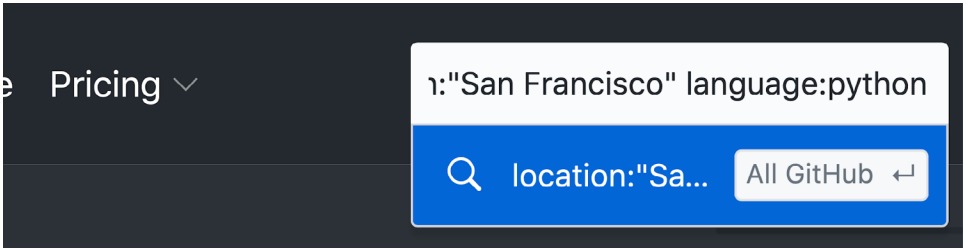


However, one main user profile element is missing when you're not logged in: member emails. Now, had you logged in first...

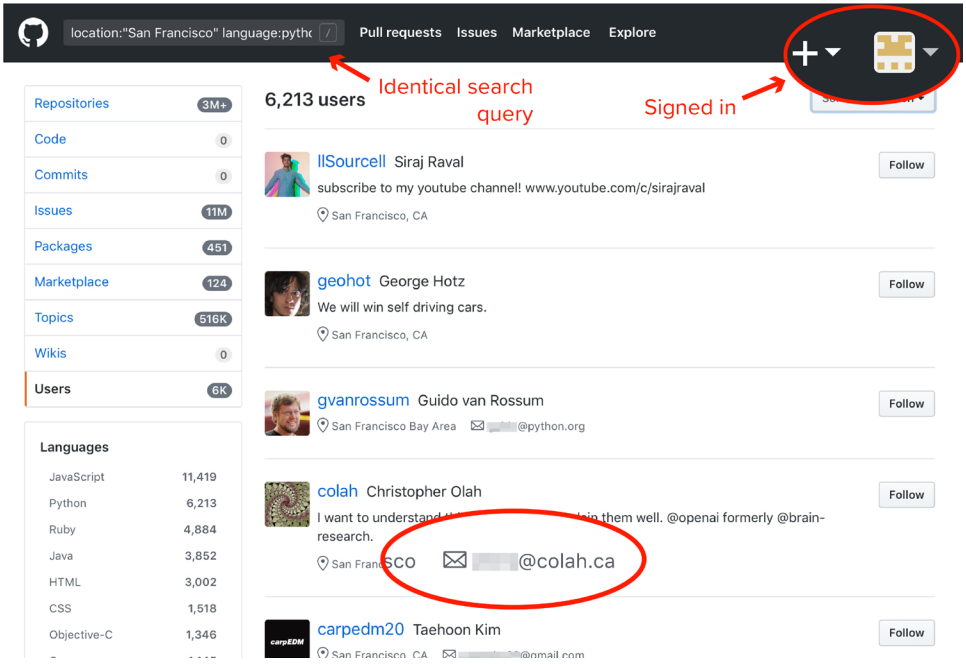


Back to top

...and then entered a search (e.g. location:“san francisco” language:python)...



...you are now able to see emails that were not visible before you logged on.



Don’t overlook this simple way of finding emails, as your sourcing time is very valuable. Start harvesting emails immediately so you can populate your CRM/ATS for the long run *while these emails are still visible on GitHub.*

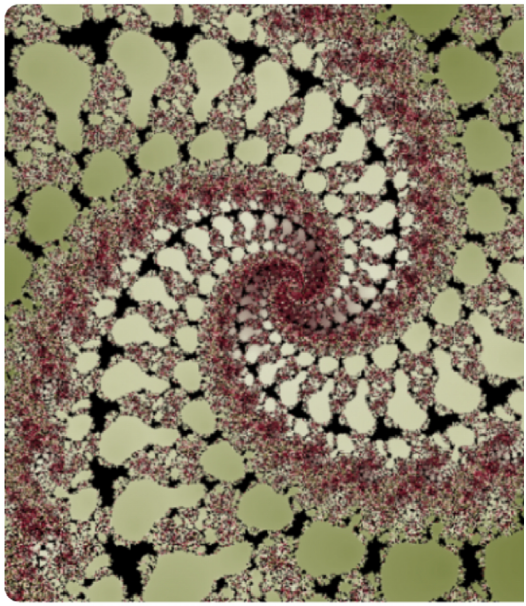
Since a member can turn this specific privacy setting off

Back to top

at any time, and with GitHub becoming more and more popular with tech recruiters who in turn will be sending more emails to those previously not barraged with messages, the time to act is now.

Of course there are numerous third party browser extensions and plug-ins, a cumbersome GitHub API hack (covered later in the ‘Advanced Searches’ section) and other round-about ways to find emails, but the most straightforward and efficient way to view and capture publicly available member emails on GitHub is to simply be logged in.

Back to top



Christopher Olah

colah

Follow

★ PRO

I want to understand things cle
and explain them well. @opena
formerly @brain-research.

San Francisco

@colah.ca

colah.github.io

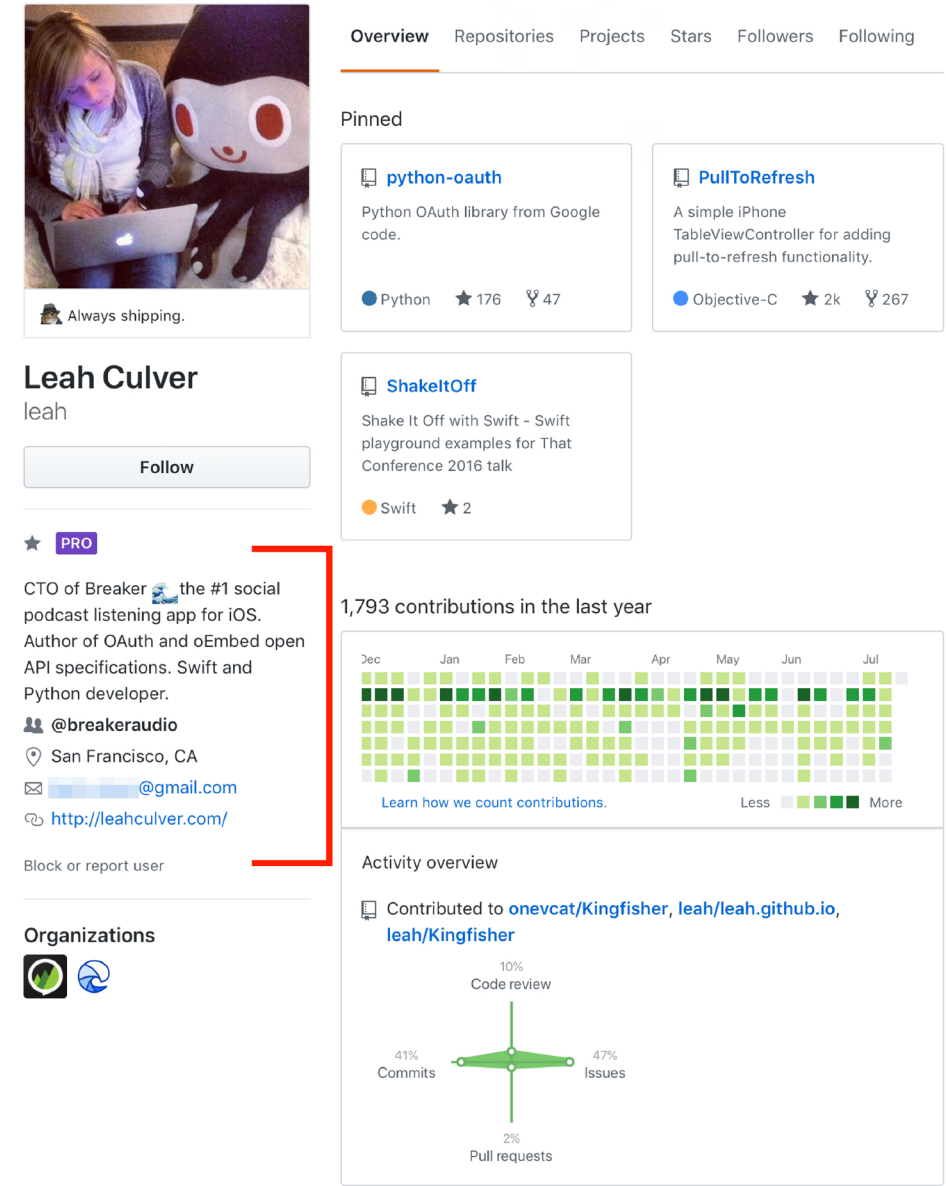
GitHub Profiles:

GitHub profiles have a standard overall format, but personal data on the left side of the profile layout can vary a lot depending on what each individual member chooses to publicly share.

The sample profile to the left provides minimal information, but does have an email listed.

Back to top

The next profile is more robust, but keep in mind that each profile tends to be slightly different:



Back to top

Each profile has two main areas:

Personal Details are housed on the left; this portion is static and never changes. Even as you choose the various navigation options along the top of the profile page (e.g. Overview, Repositories, Projects, etc.). The left side doesn't change and is always visible.

Members are not required to volunteer much information, but there is always some sort of image, with the member's nickname/username displayed below.

Keep in mind that users don't have to list their actual names or really all that much at all in their profile setups, which sometimes makes things just a bit harder for recruiters.

However, since GitHub is a social and community based platform, many developers do indeed share enough personal detail to make it a worthwhile recruiting venue. It's really nice when the members offer up valuable personalized summaries in this area (e.g. "Love surfing and cutting code in Python"), which helps you figure out what they like to do technically, what languages they prefer to code in, and so on.

Users might even share their current employer, their location, an email address and other information like blogs, links to personal projects, personal websites, or perhaps even a website they've created to promote their consulting gigs.

Back to top

These additional personal profile links can help you learn plenty about their work and personal interests, which in turn makes it easier for you to effectively personalize your messages. Moreover, if no email is available, some of these other links might open a roundabout path to contact them (e.g. contact form, an email, etc.) Be resourceful and think creatively.

Back to top

The GitHub profile URL:

One important aspect of each member profile page is the URL that is unique to each GitHub profile, which is valuable for current and future reference.

We suggest you enter that URL into your CRM/ATS for simple recall and reference: If your candidate CRM/ ATS does not already have a field designated to GitHub profiles, we suggest you add a URL-capable type data field so the link is instantly loaded when clicked on.

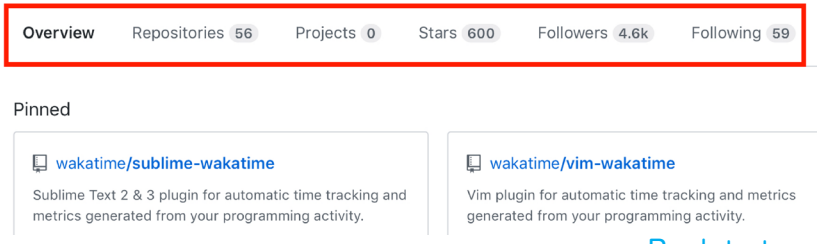
Each individual GitHub member profile URL is formatted as follows:

https://github.com/**xxx**

xxx is the unique GitHub username.

GitHub profile layout and navigation:

As you can see from the following image, all GitHub profiles have the same navigational menu choices: Overview (default), Repositories, Projects, Stars, Followers and Following.



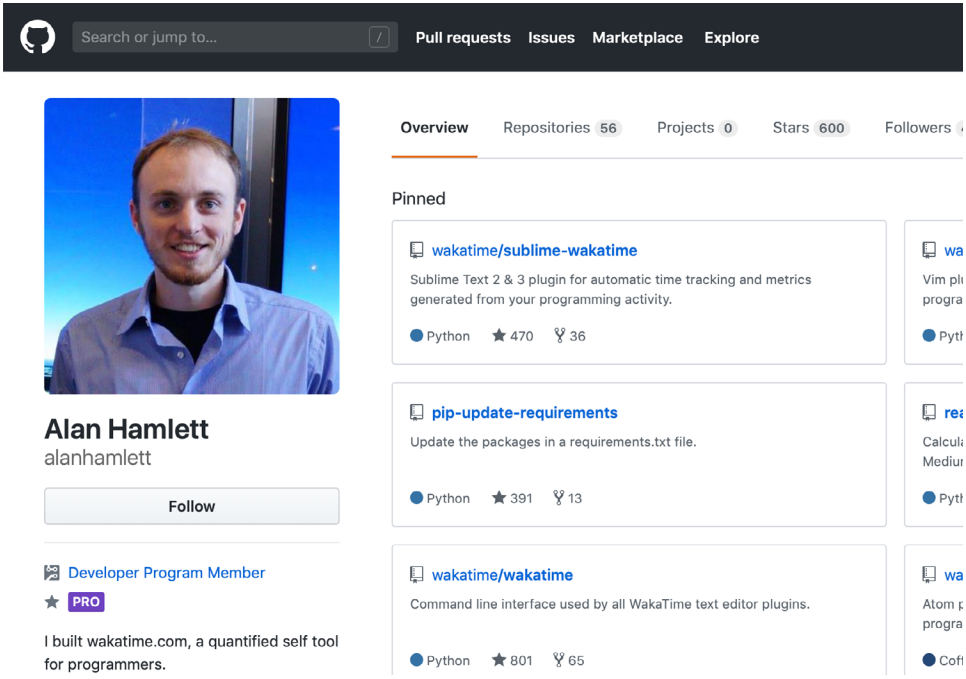
Back to top

Keep in mind that GitHub's dashboard-level, at-a-glance summary offers valuable member insights that you'd be hard pressed to uncover on LinkedIn (or Dice, Indeed, etc.), and hence the real power of being on GitHub is once again made blatant.

The more familiar you get with GitHub's profile structure over time, the more efficient you'll become in sorting through profiles. You'll become better equipped to personalize and customize messaging, quickly size up member contributions and activities, and assess user popularity among peers via a member's following and starred work.

Back to top

Overview:



This is the profile's landing page and default view. It's an efficient at-a-glance view of the developer that allows your sourcing time to be well spent. Near the top you'll see 'pinned' items. Users tend to pin repositories to their profile so other people can quickly see their best work.

Developers can also pin what are called 'gists,' which can best be described as repositories of sort. Think of gists as scratchpads or notepads used to write down small (or not so small) snippets of code. They're essentially files to be shared or information to be presented proudly to others in the community.

The presence of gists is a reminder that GitHub is for developers, and the goofy term is an example of the

Back to top

GitHub lexicon that seems to scare most recruiters off.

Don’t let these terms throw you off because it really does not matter if you know what a gist is. The goal is to get valuable actionable data that you can utilize. Explore each nook and cranny and you’ll be rewarded.

Back to top

The ‘Contributions’ Section:

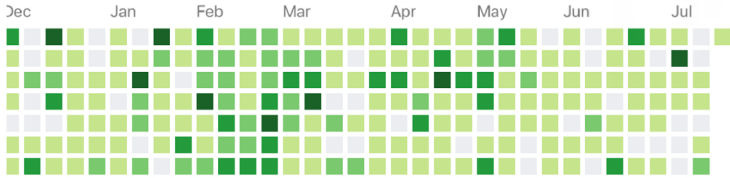
In the lower portion of the ‘Overview’ page view, there’s an area dedicated to the developer’s ‘contribution’ activity. You can refer to the image on the next page to familiarize yourself with this section.

Additionally, there is a visual graph called the ‘contributions calendar’ which shows contribution frequency over the past year. This section highlights developer contributions made to ‘other’ projects on the site, which is important because you can gain a lot of insight from exploring the stats.

As a recruiter, the odds are high that you simply don’t understand the ins and outs of coding, but as you poke around you can see keywords that help you figure out what language they’re using.

Back to top

3,067 contributions in the last year



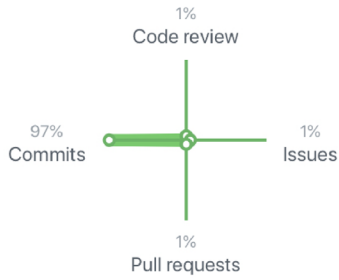
Learn how we count contributions.

Less More

@wakatime @dominino @public-apis More

Activity overview

Contributed to wakatime/vscode-wakatime, wakatime/wakatime, wakatime/sublime-wakatime and 5 other repositories



Contribution activity

July 2019

Created 45 commits in 7 repositories

Opened 1 pull request in 1 repository

vanesa/kid-o 1 merged

Support for Heroku and Netlify Jul 8

Reviewed 2 pull requests in 2 repositories

wakatime/texstudio-wakatime 1 pull request

WhatNodyn/kakoune-wakatime 1 pull request

Back to top

Something to keep in mind when looking at data on profiles:

Since users can make contributions private and/or public, keep in mind that you are only seeing what is made public. For example, contributions could show a lot less activity if they choose to hide some or all contributions made. In fact, of the 100 million repositories the platform hosts, only 28 million are made public. The takeaway is to not always draw final conclusions just because certain metrics are not visible.

Repositories:

Often called ‘repos’ or a ‘repo’, GitHub defines a repository as “the most basic element of GitHub.”

Just imagine them as a project’s folder. A repository contains all of the project files (including documentation), and stores each file’s revision history. Repositories can have multiple collaborators and can be either public or private.

One way to assess the popularity of a developer’s work on GitHub is to see how many other members have starred or forked their repos. A ‘star’ is a sign that other developers deem their code valuable. When a developer has their work forked (copied) and/or starred by other developers, this indicates that the developer has created (or is currently creating) a popular project.

On any repository page, you can clearly see how many

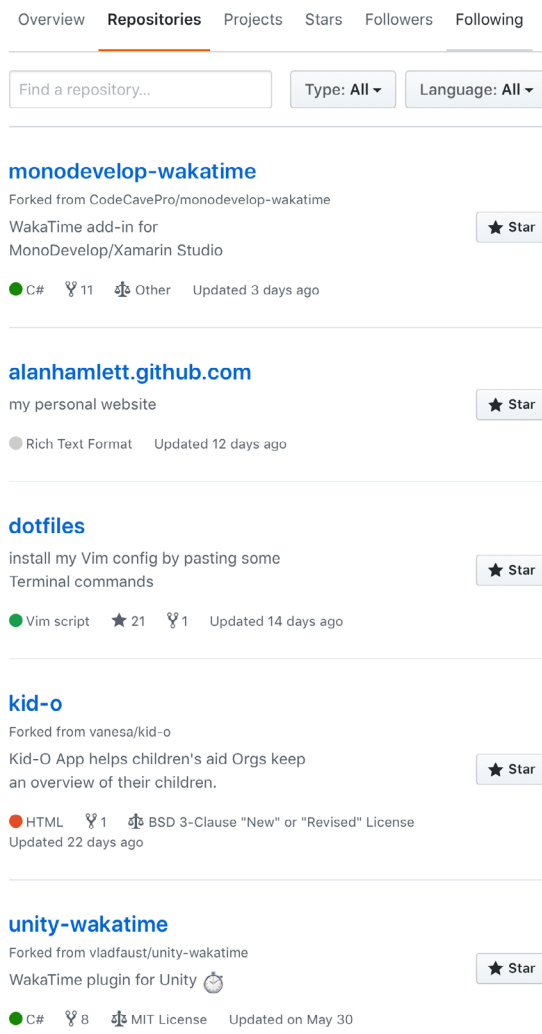
Back to top

other members have starred and forked said repository, which is good for you to know when looking for solid talent.

The act of forking allows other developers to make a copy for themselves, make changes to the source code, and use it in their own projects. This is one of the main reasons why GitHub's collaborative framework has become so popular within the coding community—and why you should be there too.

Projects:

Many confuse repositories with projects, but they are not synonymous. The 'projects' page should be thought of as a "project board" that helps to organize and prioritize one's work. All project code is hosted in repositories. This section may often appear blank, but you shouldn't necessarily worry about this.



Stars:

As mentioned above, a star is a strong sign

Back to top

that other developers consider a repository useful or otherwise valuable. In addition to rewarding good content in a similar fashion to ‘likes’ on social media, stars are a great way to mark repositories for future reference.

Followers:

Since the number of followers is a general indicator of popularity and what a developer’s reputation and perceived skill level is according to other developers, it’s a valuable piece of data for any recruiter. Simply put, it’s what the collective community thinks of one another (perceptions accurate or not).

Following:

Who a developer follows can tell you a lot, and those being followed are likely worth your checking out for good measure.

Just like every recruiter has their own approach to reviewing resumes, with social media platforms, you’ll undoubtedly develop your own systems and best practices for sizing up talent.

Back to top

Note about the profile ‘Follow’ button:

Consider using the ‘follow’ feature (via various page views in GitHub) as a way to bookmark profiles you want to explore a bit more in the future. You’d be correct in anticipating that as more recruiters discover GitHub, follower counts will saturate, but for the time being it’s still a worthwhile practice to save profiles for later by following them.

Back to top

Table of contents

Basic Searches

There are many different search 'qualifiers', as they are called on GitHub, made available; most are designed for developers to look up specific nerdy information that is germane to what most members do: write and review code.

Therefore, aside from the array of search qualifiers covered on GitHub's help section (we suggest you review and digest as you get more comfortable with GitHub search fundamentals), there are some basic qualifiers that all tech recruiters should know and put to work daily that cover most sourcing situations.

Listed with sample search syntaxes, these are:

Location:

location:“san francisco”

Language:

language:python

Back to top

Followers:

followers:>10

The ‘location’ qualifier is pretty self-explanatory. The key is to use quotes around any location that has more than one word, such as San Francisco, San Diego, or New Jersey. GitHub does not like ‘white space,’ which is any space between words.

The ‘language’ qualifier searches for users based on the language(s) of the repositories they own.

The ‘followers’ qualifier is part of what makes GitHub uniquely helpful in sizing up candidates. Followers on GitHub are a pretty reliable high-level indicator of how well the developer is regarded by the GitHub community, and can hence serve testament to their credibility.

As a general rule of thumb, GitHub members with...

- **Up to 10 followers:** these are likely good developers, and are worth learning more about.

- **11-25 followers:** these are great developers with demonstrated credibility.

- **25-75 followers:** you can expect developers with dozens of followers to be exceptional, as their demonstrated skill has allowed them to become leaders

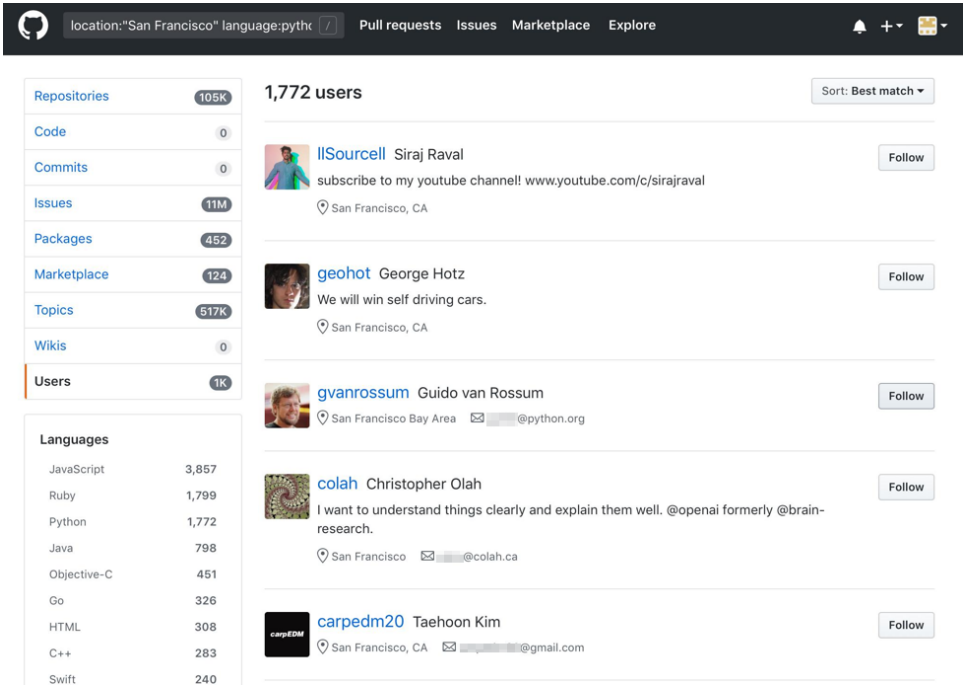
Back to top

on the platform.

- **>75 followers:** these are your coding rock stars. It might prove tricky to recruit these people.

Adjusting the ‘followers’ qualifier value is key when performing different tech stack searches. That is, if you’re trying to identify developers with specific programming language chops, and it’s a common programming language (like Python, JavaScript, HTML, CSS), increase the followers number to reduce the result set. If a language is not as popular, like ActionScript, decrease the followers number to increase the result set.

Back to top



Note that the addition of the followers qualifier narrowed down the results from 6213 to 1772 users.

Combining search qualifiers in one search string:

Know that you can combine multiple search qualifiers at once in GitHub's search box for a more refined result. For example, using the three most popular search options above, the combined query would look like this:

location:"san francisco" language:python followers:>10

This tip should cut down any confusing results sets that

Back to top

sometimes show up when organizations are mixed up with users.

As time goes on and you get more comfortable with the basics, explore search qualifiers like these to further refine your search results:

Full Name:

First Last

This qualifier is great for when you already know somebody's full name.

Repos:

>10

Great for targeting developers based on the number of repositories.

Stars:

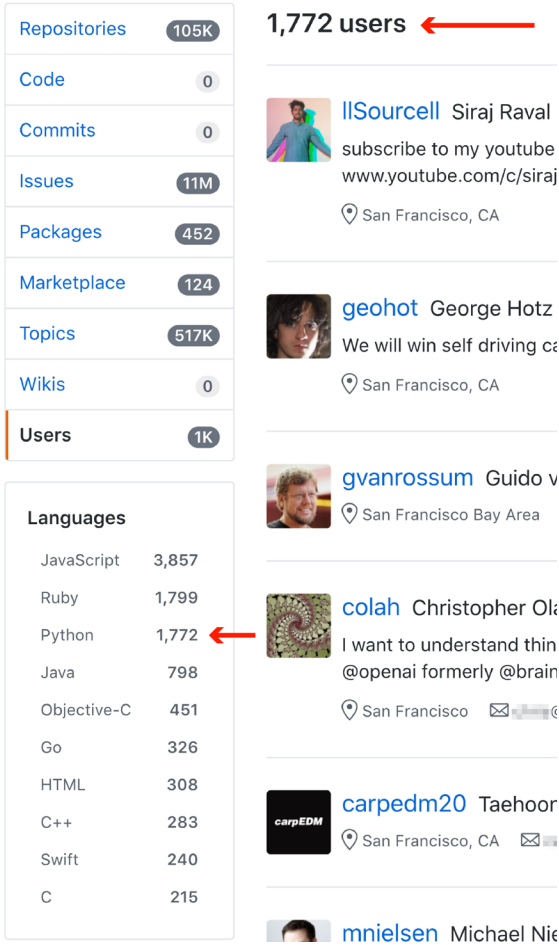
>100

Great for targeting developers based on the number of stars.

Back to top

Refining the results via the ‘languages’ section:

On the left side of the results page, there’s a section dedicated to different languages. Think of this area as a way to quickly segment or cross-reference other languages used by developers on GitHub who also meet the other search criteria like 'Location' and 'Followers'.



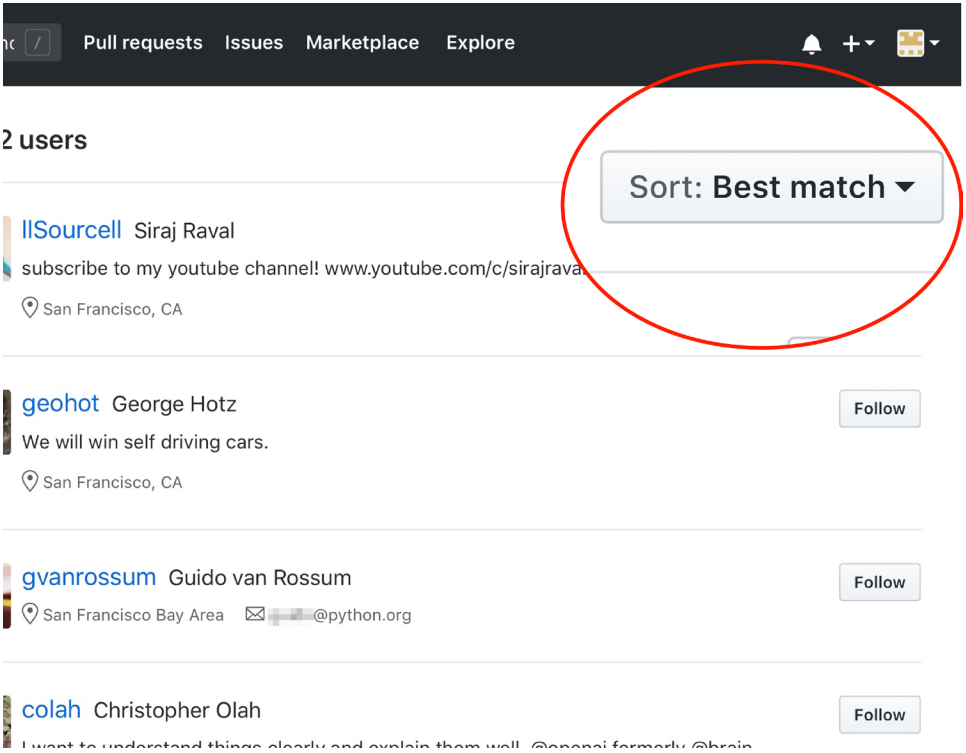
By clicking each language, you get a new result set on the right while preserving the initial search qualifiers that you typed in. Just keep in mind that the syntax in the top left search box is not dynamically changed when you do this. That is, the language qualifier syntax still reads *language:python*. Not intuitive at all, but think of the act of clicking different languages as producing different subsets/cross-sets of the initial search, independent of the language value originally typed in. See the above image to see how the number of python users in your search result matches the quantity of python users in the languages section. Experiment with this feature to see the

Back to top

possibilities.

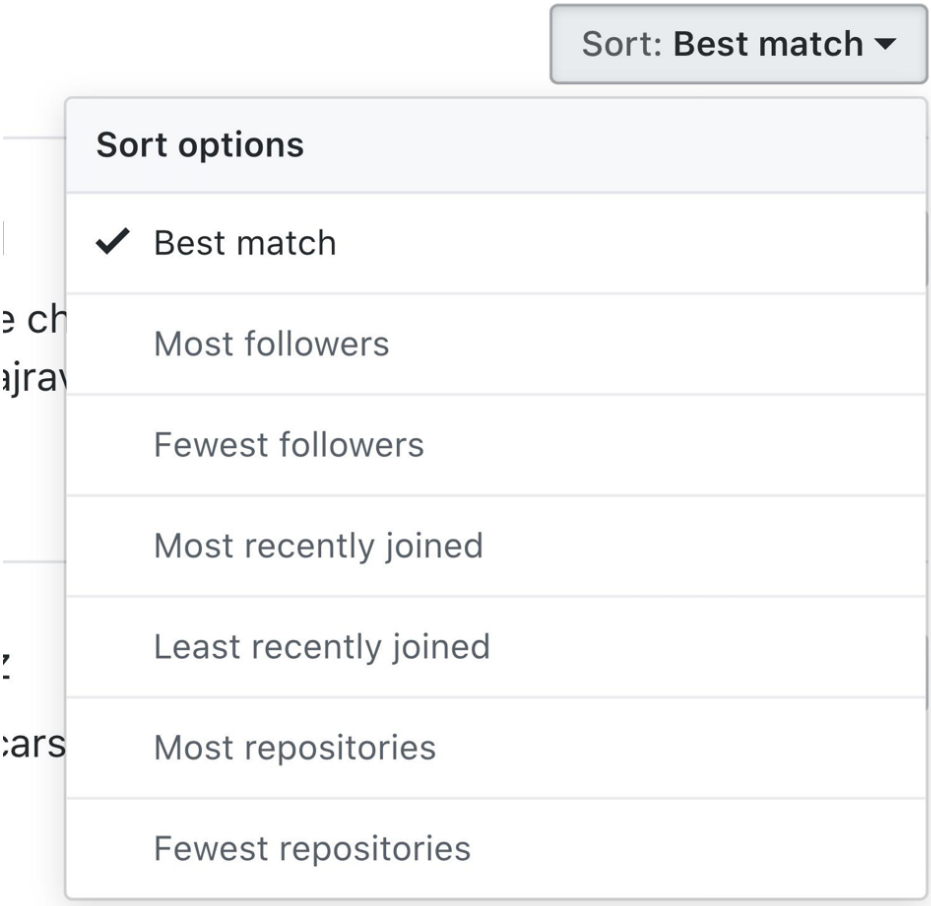
As you can see in the following screenshot, the default order of all GitHub search results are presented in what GitHub's internal methodology considers a 'Best match'.

Keep in mind that GitHub's 'best match' presentation will order developers according to how much each has coded in the specific language you've specified in your search criteria near the location you've included. This default result view makes it easy to find people whose primary coding language is the one you've clearly chosen to search for.



Back to top

If you click on the drop down...



...you'll see that you have all types of ways to sort result sets from top to bottom. These other sorting options are important because the 'best match' results are often the developers that many other recruiters have attempted contact with. Therefore, there's a chance they won't give you the time of day.

Back to top

If you’re going to use the best match choice, perhaps start your search three, four, five, or ten pages down as many other recruiters will not search that far. Just a suggestion to help you separate yourself from the recruiting masses.

Back to top

Table of contents

Advanced Searches

Before we cover GitHub's Advanced Search feature, let's go over a method that often uncovers a GitHub member's email address if it's not already publicly visible on their profile. Some refer to this back door approach as the "GitHub API hack." It's cumbersome, but effective in a pinch. Follow these steps:

- The first step is to capture the GitHub member profile username. Remember that when you're sitting on a user's default overview profile page, the username is always the last portion of the profile web address as follows: https://github.com/xxx

- Make a note of the Username portion from the URL above—perhaps paste it in a virtual notepad.

- Copy and paste the following URL into a new browser tab (we suggest you bookmark this address for easy future use): https://api.github.com/users/XXX/events/public - but don't expect any results after hitting enter as this is not a valid page.

- Next, replace XXX with the username and hit enter.

- The screen will fill up with code.

Back to top

- Use the browser's 'Find' command and search for the keyword 'email' and most of the time the member email will be disclosed.

Above you'll see that this advanced trick uncovers an email address not visible on this particular developer's profile page.

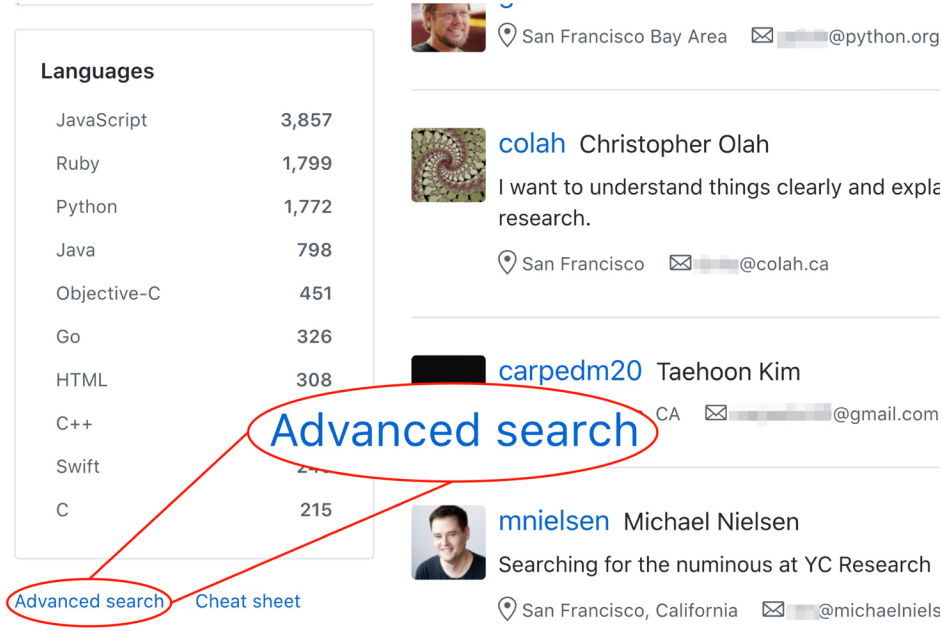
Back to top

Using GitHub's 'Advanced Search' feature:

There are two main ways to search for users. You can use the free-form search box approach covered in the previous section (in which you have to manually type in search qualifiers), or you can use GitHub's advanced search feature that helps you build search strings in a wizard-like way.

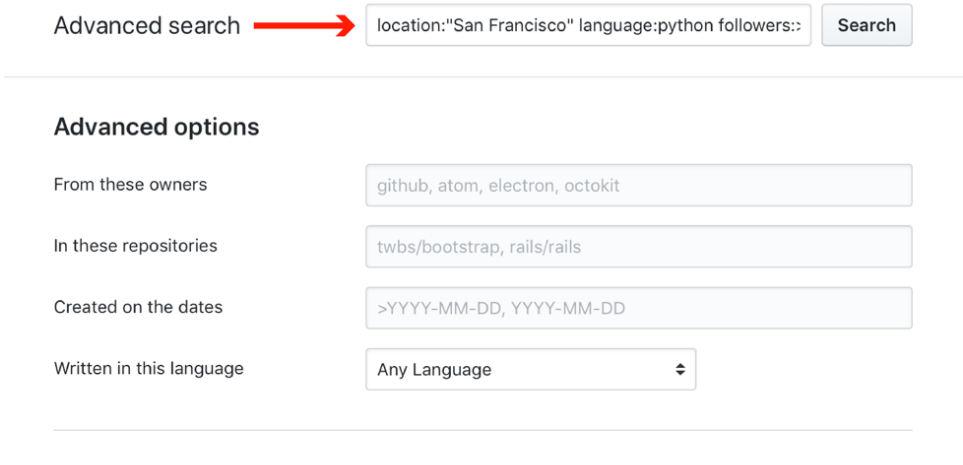
Many recruiters prefer the advanced search approach so they don't have to remember exact search qualifier syntax or risk data entry typos. However, there are pros and cons to each approach, so be sure to utilize both options to figure out what works best for you.

Back to top



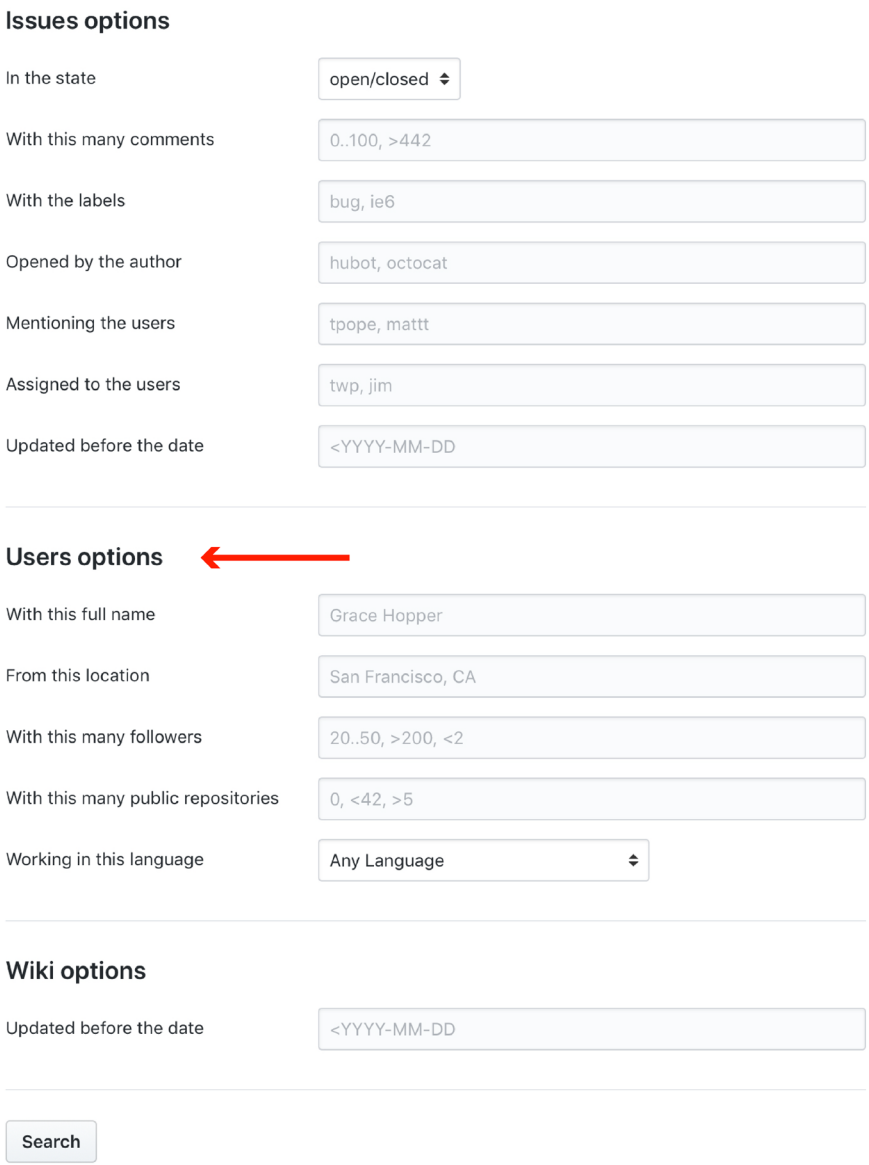
Clicking on this link from the search results set page takes you to the built-in GitHub advanced search page.

From our previous example search, the top of the advanced search page looks like this:



Back to top

As you choose options below (you'll generally want to focus on the 'Users options' section near the bottom), that search box will be updated.



Back to top

The search box syntax at the top will continually update the search string you started with as you make choices in the many fields below. Once again, most of your searches with this tool will be from the ‘Users options’ area near the bottom.

If you access the search page directly—not from the ‘advanced search’ link on the search results page—the search field is blank. Think of it as a clean canvas to craft a brand new search. Once again, as you choose options, the field up top is manipulated.

Tip: GitHub does not save searches and related syntax (other than via the use of the ‘previous page’ arrow feature of your browser) so be sure to cut/paste the ones that you find helpful and create a list for future use.

Back to top

Table of contents

How to Evaluate Developers on GitHub

GitHub can actually help technical recruiters effectively assess candidates both technically as well as a behaviorally (e.g. soft skills) which are both extremely valuable. Let's look at the technical assessment first.

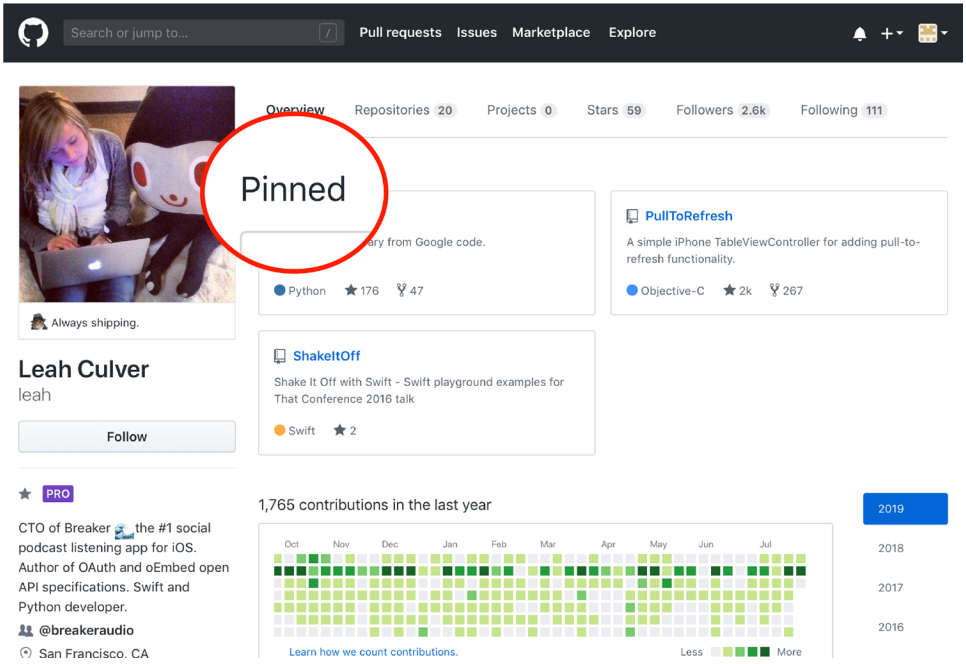
Technical assessment:

Since much of the code written by developers on GitHub is made publicly available for review and evaluation, GitHub should not be overlooked by tech recruiters, as no other platform makes this possible. Also, compared to LinkedIn's 'Skills & Endorsements' profile feature, GitHub is showing you specifically what a user can code in, which in most cases is much more valuable to a recruiter than keywords on LinkedIn.

However, the reality is that most tech recruiters don't have any coding experience, so making sense out of all the source code is a legitimate concern at first glance. Nonetheless, you'll soon discover it's actually quite painless to size up a member's technical focus using GitHub repos.

Back to top

Languages used in member Repositories:



‘Pinned’ repos on the user’s default ‘Overview’ page act as a sort of ‘greatest hits’ chosen by the profile owner. Each repo summary will display (in most cases) a programming language (as highlighted in the image above), which makes it easy for a tech recruiter to tell what language was used to produce that specific repo’s code. With a quick scan of all the pinned repos, you’ll notice one of two things: most are written in one primary language (e.g. six of six are written in Python implying that they’re probably a Python coder) or the array of pinned repos show various languages listed for each.

In either case, your next move is one click away and should not be overlooked: simply click on the Repositories tab at the top. We highly recommend looking at the

Back to top

member's entire portfolio of repos to see what they tend to code in the most or to see how wide their skills range. Browsing their entire collection of repos and looking for the programming language listed for each (every now and then there is no language listed), you not only see the breadth of languages they utilize, but it can also be a way to validate if they are a hard core coder in one or very few specialized languages.

Of course there's no guarantee that these are the languages they are the most proficient in professionally (as lots of GitHub members use the platform to do things they don't get to do at work, are curious about, etc.), but this assessment provides greater insight into their coding passions, which is great data for crafting effective messaging that might get them to reply to you.

In the case above, Objective-C and Swift (both iOS focused type technologies) may indeed allow you to draw a conclusion at the pinned repo level. However, viewing all the repos under the Repositories tab tells you that they do more than just Swift and Objective-C programming. This user has also done work with Ruby, CSS, JavaScript, HTML, and Python. No matter how compelling the pinned repo list is, we encourage you to always browse the repos tab for the most comprehensive sampling so your assessment is an informed one.

Keep in mind that often a member tells you exactly what they do and what they love by including this information in their personal profile summaries. That is, users often share what their preferred languages are, and in some cases, what their passions are. However, using both personal

Back to top

profile summaries and the repo tab contents is the best combo for comprehensive assessment.

Tip: If you work as a corporate recruiter and have access to the dev team or specific dev team members or managers, you can always share GitHub profiles and/or repos that you want reviewed. As a reminder, the unique GitHub profile URL looks like this:

https://github.com/xxx

...where xxx is the 'unique' GitHub username.

For a specific repo that you may want reviewed, the URL looks like this:

https://github.com/xx/yyy

...where xxx is the username and yyy is the repository name.

Simply copy, bookmark, cut/paste, etc. so that you can pass on the link to one of your company's dev team members.

What heavy Forking activity can tell us:

If a developer on GitHub comes across code written by someone else that could be of use to their own projects, they have the option to fork it for their own use.

Back to top

In essence, forking is when code gets copied for use in projects it was not originally written for. How often your own work is forked is a good sign that the GitHub member's code is valued by others. If a member's code shows a high count or frequency of being forked, this is one indication that they may know what they are doing.

Behavioral/soft skills assessment:

Techie soft skills and behavioral traits are something a recruiter is continually asked to size up by their clients and/or internal hiring managers. Even though LinkedIn has a 'Recommendations' testimonial type feature in an attempt to endorse and describe personal characteristics, it tends to be biased simply by it's non objective nature.

However, since GitHub is a developer's natural habitat where they are just being themselves, their ongoing community based 'activity' (e.g. Pull Requests, Forking, Accepted Pull Requests, etc.) often reveal behavioral traits, which allows a recruiter to draw pseudo educated conclusions on their soft skills.

Back to top

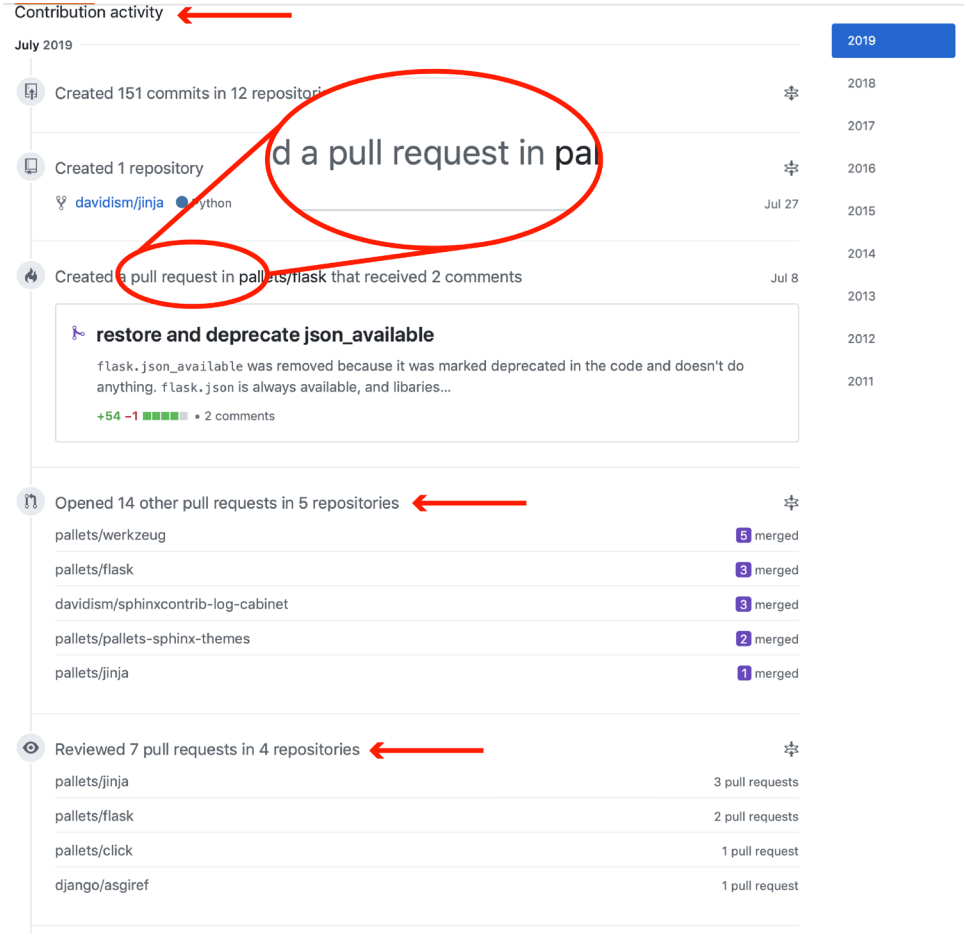
What can a recruiter learn and potentially conclude from the following user activities?

‘Pull Requests’ are requests made to the original code’s author that changes by the requesting developer be considered. They are essentially one user who wants to fix bugs, add new features or change features, and then hand the adjusted code back. The developer submitting the pull request can also add notes and related comments about the changes rather than just submitting code blindly.

A high number pull requests is a pretty good sign that a member has a genuine passion for writing code, is collaborative in nature and also likes contributing to the GitHub community. It’s really not far fetched for a recruiter to conclude with some degree of certainty that actions like these reveal a developer’s desire to collaborate with other developers, and that they’re open to putting themselves in a vulnerable position (e.g. pull requests being rejected). These traits might indicate a developer to be a good team player.

Pull requests are scattered throughout individual repositories, but for simplicity, while on the user’s profile in the default ‘Overview’ page mode, it’s best to just scroll down past the contributions calendar graph and quickly scan for pull requests. The screenshot on the next page highlights what you should be looking for:

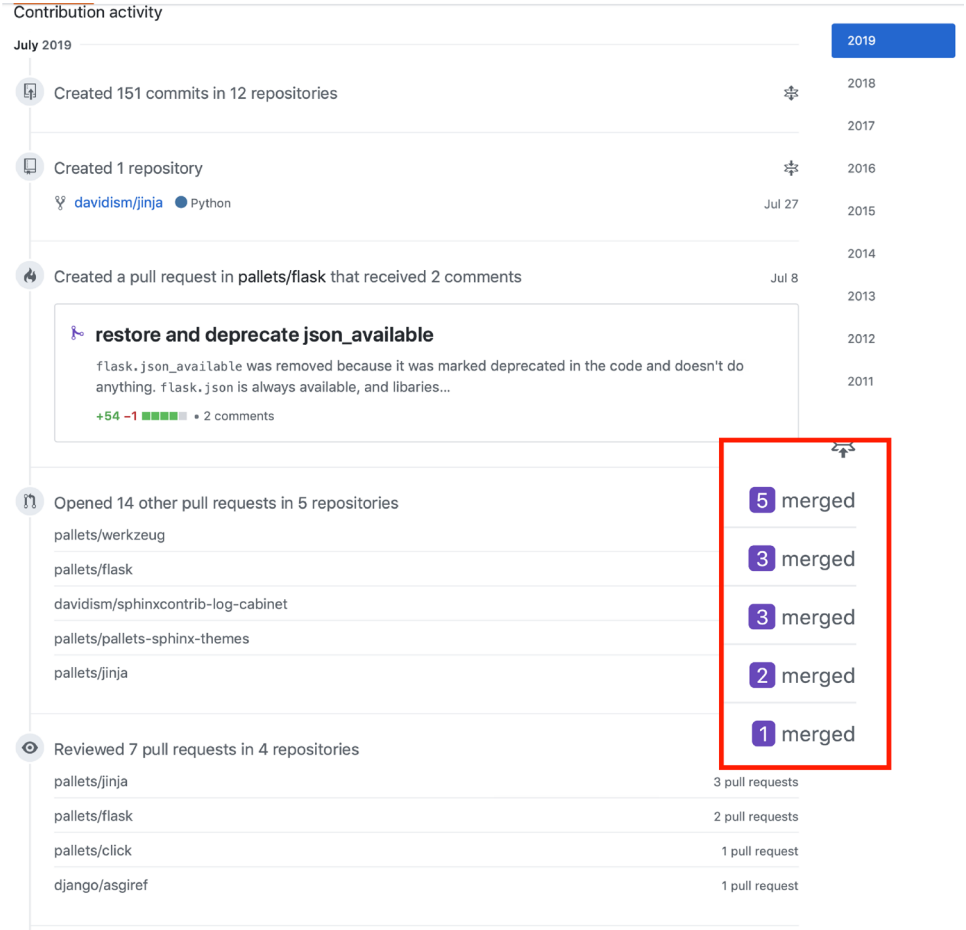
Back to top



‘Accepted’ Pull Requests are the code changes submitted by one developer and accepted (or merged) by the original code author, which indicates that the changes are valued. An accepted pull request is potentially a good sign that the requester is a decent coder.

As you can see from the following screenshot, after expanding the pull requests via one click, you expose the status of the pull requests. If the word “merged” is present, that means it was ‘accepted’):

Back to top



What does this tell you about the accepting party? In this case it's a possible indication that they are not a diva, they're receptive to constructive criticism, and that others have talents that they can appreciate and that they might be coachable. It's not that much of a stretch to consider these as signs that they are wired as a team player.

Forking is the act of copying another developer's code for one's own projects. The person who forks another developer's code might signal that they are curious and open to what others do, and not just focused on their own coding universe and technical ego. It's a lot like a recruiter who reads other recruiter blogs in an effort to up their

Back to top

recruiting game, which ultimately makes any professional even more effective in their career.

Sometimes when a developer on GitHub forks somebody else's code, they'll then make a Pull Request (covered above in this section) to ask that suggested changes and enhancements to the code be considered by the repo owner. As a result of this interaction, it would not be far fetched to attribute a developer's frequent submission of pull requests to a naturally collaborative, team-oriented attitude.

Participation levels on GitHub:

Previously covered in the 'Tour of GitHub' section, 'contributions' on GitHub are one way of analyzing and gauging participation levels. In general, a high volume of contributions and activity is a good thing. With the dedicated contributions summary section on any given member profile, and the related contributions calendar graph, you do indeed get a good feel for a member's overall level of activity. However, like anything else, contribution levels alone need to be weighed in context to all other information you've gathered.

For example, don't just assume that a low contribution level is a valid sign that somebody is not a formidable coder.

Furthermore, someone with a lot of activity might be more engaged simply because a lot of their code on GitHub is related to work projects. Naturally, these people will log more activity than the average coding enthusiast that is

Back to top

mainly working on projects in their spare time.

You also don't know anything about a user's personal life that might affect activity levels on GitHub. For all you know, a person might have six kids or travel a lot, so less activity shouldn't necessarily falter your interest right away.

The nice thing about the contributions calendar view is that if somebody has peaks and valleys of activity, you might be able to surmise patterns that may reflect their work ethic. Maybe one potential candidate is a weekend coding warrior. Perhaps another is the opposite, coding on weeknights and mountain biking or playing D&D on the weekends. Just know there could be more to the story than meets the eye upon first glance.

As a way to assess the passion of developers, tech recruiters often ask potential candidates what they do on their time off. As most tech recruiters know, this is by no means a trivial question. Do they completely check out of work because coding is just a way to pay the bills, or are they always working on something to satiate their hunger for solving problems and doing new and neat things? Read between the lines in regard to profiles with less active contributions levels because you don't want to throw the baby out with the bath water based on a single metric.

Back to top

Table of contents

Challenges with GitHub

GitHub obviously brings a lot of sourcing related value to the table and provides information that you won't find in many other places. Where else can you view actual code written by potential development candidates which provides insight from an angle a resume simply can't offer?

Also, you're often looking at developers that are simply not present or active anywhere else which is one of the main reasons to source on GitHub. These unique platform benefits are part of the reason that GitHub must be part of your sourcing strategy moving forward if not already.

However, GitHub also has its share of limitations and challenges from a recruiting perspective.

Profiles Missing Key Data:

Many members have sparsely populated member profiles where personal data is simply not listed or made available to the public. More specifically, they often don't provide the following elements in their summaries, which can be frustrating when sourcing for technical talent:

Back to top

- 95% are missing social media or blog links
- 94% are missing an Email
- 92% are missing a company name
- 90% are missing a Location
- 85% are missing Full Name

Focusing on location alone, which is undoubtedly a key query element of the majority of searches conducted by tech recruiters, if GitHub searches include the location qualifier (e.g. location:“san francisco”) you’ve potentially missed 90% of the members on GitHub that might live in that target location.

With the statistics above in mind, even if you find a developer on GitHub with the skills and experience you require, it’s highly likely that their profile will not have all the information you need.

What does GitHub’s ‘Best Match’ default search filter mean?

Well, your guess is as good as anybody else’s. How GitHub ranks your search results by ‘best match’ is not public knowledge, and is difficult to theorize.

While you can search by members with specific numbers of stars or followers, results are not necessarily ranked

Back to top

in order of technical proficiency. Also the total number of stars and followers may not give you the best developer in a certain technology.

More specifically, Developer A could have 50 stars or 50 followers but they only code in Python versus Developer B who has 60 stars or 60 followers but their repos are spread out with numerous technologies involved.

You can also search/sort repos with the most number of stars, but again this is potentially problematic because...

1. It is only one repo of someone's portfolio. Having a way to analyze the popularity of repos grouped together by technology would be a far better method of evaluating talent. It's like trying to evaluate a musician's by listening to their top hits versus their entire body of work.

2. Sorting by repos on GitHub is also painful as chances are all the top repos are by ORGs and not individual Users. Even when you find one belonging to users, often they are either in the wrong location or their location is missing.

Profiles not always up-to-date:

A profile that appears complete won't necessarily be up-to-date. In fact, 38% of GitHub members have never updated their profiles and the average number of days since a member last made any changes is 252. Furthermore, there's no practical way of seeing when an

Back to top

individual profile was updated. However, to help remedy this reality, it's good to keep in mind that profiles active with recent commits (which are changes to one or more files and similar to saving a file) are more likely to have a more updated profile as a result of it.

Misleading profile stats:

If you land on a profile with a low amount of stars or followers, it does not necessarily mean that the developer is bad. That is, a seasoned developer with solid programming skills might not spend a ton of time on GitHub and hence their profile presents a smaller body of work than they've truly produced. Or they are very active on GitHub but they have fewer publicly visible repos available.

Missing developers:

Obviously not every member on GitHub is a software developer nor is every software developer on GitHub. Realistically there will always be a mix of developers that are: only on GitHub, only on Stack Overflow, on both of these popular developer-centric platforms, on GitHub, Stack Overflow and LinkedIn, and some only on LinkedIn. There are also similar sites to GitHub like Bitbucket and GitLab that some developers prefer and are loyal to.

Time will tell if GitHub is the ultimate destination for years to come or if it's just another emerging platform that is the current mecca for developers to flock to. For now, GitHub (and even Stack Overflow) are great places to fish while the fishing is good and the word is still spreading.

Back to top

However, just like our recommendation about LinkedIn not being your sole source, GitHub is but one important tool for you, but by no means the end all source to use for tech recruiting.

Search queries must be precise:

GIGO stands for 'Garbage In, Garbage Out'. Nothing could be more true when it comes to GitHub search syntax. It's one reason many recruiters might want to use the built-in 'advanced search' feature. Just know that you should always double check every character and space of your syntax if things don't look right. Since GitHub will not let you know if you type search terms in wrong, we suggest you save your popular search syntax queries for cut/paste application to reduce inaccurate results.

Profiles with minimal information:

Knowing GitHub members have the choice to make their work public or keep it private, don't immediately jump to a conclusion when you see a sparse profile. Less activity on a developer's profile could simply mean they don't want to make their work public. It could just as easily mean they're working on confidential projects (which is an intriguing detail about a potential candidate in itself). It could also be that they're relatively new members who haven't spent much time on the platform. Once again, everything on GitHub is about context and totality so don't base any conclusion on single metrics.

Back to top

Table of contents

Solutions to Compliment GitHub Recruiting

GitHub by itself should absolutely be an existing and essential resource for any tech recruiter. With powerful third party add-ons that can unlock GitHub's full potential, it can be effectively augmented into the killer recruiting app of the future.

As noted in our last chapter, there are indeed some clear challenges when recruiting on GitHub that a commercial tool could possibly help resolve.

GitHub's difficult and awkward non recruiter friendly user experience is why there are a number of fine tools designed to help. Sourcing expert Dean Da Costa has compiled a comprehensive list for recruiters to explore at:

https://start.me/p/EL84Km/cse-utopia

That being said, we can speak with authority only of our own solutions. Dedicated to tech recruiting exclusively, with developerDB we've taken lengths to address all the aforementioned limitations to recruiting with GitHub.

Back to top

Sourcing Solution:

Our flagship tool is a cloud-based front-end search interface that allows recruiters to mine our vast database of technical candidates.

Since the talent pool is continuously aggregated from both tech and non-tech sites, the database contains members from GitHub, Stack Overflow, LinkedIn and most sites where tech talent can be found. Since developerDB pulls from so many different sources, the database contains many hard to find candidates that other search engines can't uncover and have a more complete "master profile" of a person than any one site can provide.

Chrome Extension:

As a compliment to the sourcing tool covered above, developerDB offers a very easy to use, on-demand Chrome extension that recruiters use when they're already on a candidate's social media profile and want to unlock additional hidden information.

Since both products work in concert, there is no need to unlock profiles with one product when you've already unlocked them with the other. The Chrome extension is not only a great resource for all your known and unknown candidate profiles on GitHub, but on a handful of other popular social platforms including LinkedIn, Stack Overflow, Meetup, Twitter, and others.

The developerDB Chrome extension has a very unique

Back to top

capability that LinkedIn's Connectifier extension doesn't offer when a recruiter is viewing a GitHub repository. developerDB's solution is fully functional at the profile and the repository level. Connectifier does not function when recruiters are reviewing somebody's repositories. Connectifier users must switch back and forth from repositories to a that members profile first to engage their extension, which is very cumbersome and inefficient.

Tip: to get back to any specific repository, the URL recipe is:

https://github.com/xx/yyy

where xxx is the username and yyy is the repository name.

Back to top

Candidate ranking—a tech recruiter's best friend:

Since most tech recruiters are not inherently technical, one of the most valuable benefits is developerDB's technical ranking feature. With this feature, developerDB does the heavy lifting on sizing up technical talent, which comes in handy when you don't have access to a team of developers to bounce candidates off in the early stages of sourcing.

What effectively separates developerDB from all the other tools out there is their dedicated and relentless focus to tech recruiting and the related IT-focused talent that tech recruiters need to find. Since developerDB concentrates on building a database rich in exclusively technically focused professionals, they can be a valuable and cost effective tool for your tech recruiting arsenal.

Back to top

Table of contents

About the Authors

Alex Wenz - Alex is a veteran tech recruiter currently performing as an industry Advisor to developerDB. Originally from Long Island, New York, Alex has been living in San Diego for the past 30 years.

Don Phan – Don is the founder of developerDB. He is a data nerd as well as an award winning marketer to tech geeks. Don lives in San Jose, CA.

Jakob Picciotto - Jakob is a student at Whitman College in Walla Walla, WA, pursuing a major in Rhetoric, Writing, and Public Discourse. Jakob is an Intern for developerDB.

Back to top

Table of contents

Glossary

LinkedIn InMail: A message sent on LinkedIn to a user that the sender is not connected to. Depending on subscription level, account holders are limited to a specific number of InMail messages every month.

GitHub: The world's largest online software development community.

Repositories: GitHub defines a repository as "the most basic element of GitHub." One can imagine them as a project's folder, containing all of the project files (including documentation), and storing each file's revision history. Repositories can have multiple collaborators and can be either public or private.

Personal email address: A point of email contact other than one's work email address. This is the preferred channel to reach potential candidates, as they are more reliably consistent than work email addresses.

Search syntax: The format in which a search query is written. With GitHub, it is important that you pay careful attention to your search syntax to get the desired result.

Back to top

CRM/ATS: Both serving as digital recruiting tools, candidate relationship management systems and Applicant Tracking Systems are fundamental to any tech recruiter's workflow. Your ATS is a repository for applicants and facilitates the application process. Your CRM helps you maintain relationships with potential candidates of interest who aren't necessarily applying for any positions yet.

API: Standing for application programming interface, the API is a set of functions and other resources that allow applications to access the capabilities of other applications. In the case with this guide, you're able to hack personal email addresses by taking advantage of the API that GitHub provides to the public.

Overview: The default view when visiting a GitHub user's page. It's an efficient at-a-glance view of the developer, and has been inadvertently optimized for recruiting efficiency (lucky you!).

Projects: Not to be confused with repositories, the projects page can be thought of as a virtual project board that helps users organize and prioritize their work. All project code is hosted in repositories.

User: An individual account holder on GitHub. Nearly all of your searches performed on the platform as a recruiter will be for users. Users are distinct from **organizations**, which are accounts for corporate entities.

Back to top

Pinned: Pinned items are those that a user opts to feature prominently on their ‘Overview’ page. For recruiters, pinned repositories allow you to quickly see a potential candidate’s best work.

Followers: Much like followers on popular social media platforms, a user’s followers are people who have opted to see continual activity from said user. Followers on GitHub are a pretty reliable high-level indicator of how well the developer is regarded by the GitHub community.

Following: A user’s ‘following’ list includes all users they’ve opted to see continual activity from. Who a developer follows can tell you a lot.

Contributions: Contributions encompass all collaborative activity performed by a user, and can be easily viewed on the ‘Overview’ page.

Stars: Similar to a like on social media, stars reward users for valuable content. If a user wants to easily refer to a repository later, they can star it and find it in their ‘starred’ list.

Follow button: For recruiters, use the follow button to “bookmark” users for future reference.

Search qualifier: A criterion through which you can narrow down searches. Examples include location and number of followers.

Back to top

Best match: This is the default order of all GitHub search results. It is not known exactly how GitHub's algorithms choose which results to display first, so it's good for recruiters to be in the habit of looking past the first page of results.

Advanced search: Use advanced search to perform searches with multiple qualifiers without having to worry about perfect search syntax.

Technical assessment: A recruiter's evaluation of a GitHub profile based on gleaned technical focus, and language abilities. If you have a dev team on hand, you can take technical assessment a step further by examining potential candidates' code.

Soft skills assessment: A recruiter's evaluation of a potential candidate's personality and work ethic based on specific patterns seen in certain metrics.

Forking: When a user wants to use someone else's code for their own projects, they can fork it. Forking allows you to make copies of others' work while still giving them credit, and allowing others to easily see where it came from.

Pull requests: When a user wants to make changes to someone else's repository, they submit a pull request.

Accepted pull requests: If the original repository owner approves of additions or changes made in a pull

Back to top

request, they accept it.

Chrome extension: An application that lives in your browser, serving to enhance a specific aspect of your experience on the web.

Back to top

Table of contents

Index

Back to top

L

M

P

R

S

Back to top

T

U

Back to top