Improving your team’s source code searching capabilities

Nikos Katirtzis
Software Engineer @ Hotels.com

Who am I?

Educational Background

- 👨‍🎓 Meng in Electrical and Computer Engineering (Aristotle University of Thessaloniki)
- 👨‍🎓 MSc in Computer Science (University of Edinburgh)

Working Experience

- 👨‍💼 Software Engineer at Hotels.com (Expedia Group)
  - Part of the team that's responsible for user authentication and identification (~2 years).
  - Recently joined a team that's exploring and evaluating new technologies.

Projects/Interests

- 👨‍🔬 Developed Mantissa, a TDD code search engine, and CLAMS, an approach for mining API usage examples from client source code.
- 👨‍🔬 Particularly interested in source code searching/mining.

Presentation structure

HApiDoc

Part 1 – Searching for source code

- Why you need a source code search engine
- Overview and comparison between the most popular code search engines
- Recommendations and what you need to consider
- Recent advances

Part 2 – Searching for API usage examples

- HApiDoc: A service that mines API usage examples from client source code
- CLAMS or behind the scenes of HApiDoc

PART 1

Searching for source code

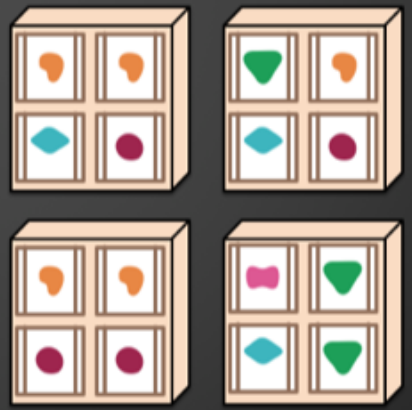
Monoliths are dead, long live microservices!

A monolithic application puts all its functionality into a single process...

... and scales by replicating the monolith on multiple servers.

A microservices architecture puts each element of functionality into a separate service...

... and scales by distributing these services across servers, replicating as needed.



Source: https://martinfowler.com/articles/microservices.html

Monoliths are dead, long live microservices?

Monoliths are dead, long live microservices.

Why you need a source code search engine?

How many searches does the average developer perform on an internal code search engine on a typical weekday?



A. 0

B. 1-2

C. 5-10

D. >10

Source: Sadowski, Caitlin, Kathryn T. Stolee, and Sebastian Elbaum. "How developers search for code: a case study." *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering*. ACM, 2015. Available at https://research.google.com/pubs/pub43835.html.

Why you need a source code search engine?

“Software engineering is more about reading code than writing it, and part of this process is finding the code that you should read”. (Han-Wen Nienhuys - author of Zoekt)

- 👍 Understand code dependencies in order to avoid breaking changes.
- 👍 Fix production issues faster by locating the root cause.
- 👍 Find references of hosts/code that will be deprecated.
- 👍 Avoid duplicating existing code.
- 👍 Share coding solutions and styles.
- 👍 Locate security problems (e.g. hardcoded keys/passwords).

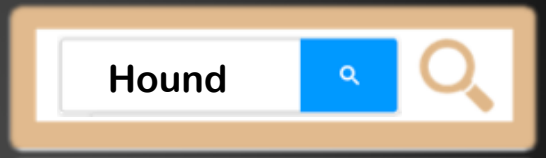
Can’t I use my Git hosting service’s search?

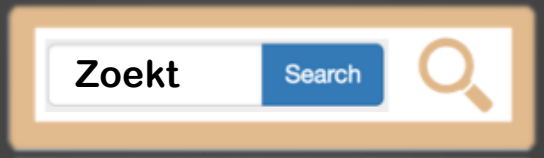
Would you go to a souvlaki shop for fish?

Can’t I use my Git hosting service’s search?

- 👎 No partial/substring matching.
- 👎 Special characters are removed before indexing and are not allowed when searching.
- 👎 Case sensitive search not possible.
- 👎 No regex.
- 👎 Cannot configure or add any features since this is a solution that's integrated into GitHub Enterprise.
- 👎 Best match is based on naive techniques (tf-idf).
- 👎 Too much unused space in the results page!

Popular CSEs



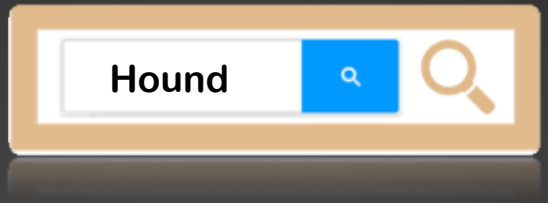


- Searchcode Server is a powerful code search engine with a sleek web user interface.
- Uses Lucene to index code and provides rich additional features.
- Developed by Ben Boyter.
- Originally a public source code search engine (searchcode.com), later the developer created Searchcode Server which is the enterprise version.

Searchcode Server screenshot

Searchcode Server pros/cons

Pros	Cons
✓ Consistent speed regardless of search	✗ Limited partial/substring matching
✓ User friendly filtering by repo/user/language	✗ Does not support case-sensitive matching
✓ Rich UI	✗ Special characters removed
✓ Relatively easy to setup, maintain and monitor	✗ Does not fully support regex
✓ APIs	✗ Inconsistent search results



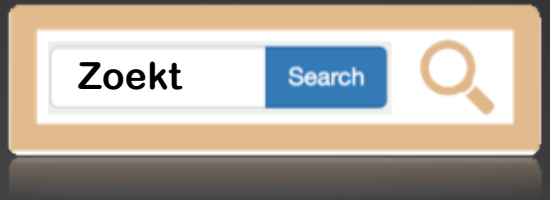
- Hound is an open-source source code search engine which uses a static React frontend that talks to a Go backend.
- Uses ngrams for indexing and matching.
- Created at Etsy by Kelly Norton and Jonathan Klein.
- Its core is based on Russ Cox's "Regular Expression Matching with a Trigram Index or How Google Code Search Worked" article and code.

Source:. Rus Cox. "Regular Expression Matching with a Trigram Index or How Google Code Search Worked." 2015. Available at https://swtch.com/~rsc/regexp/regexp4.html.

Hound screenshot

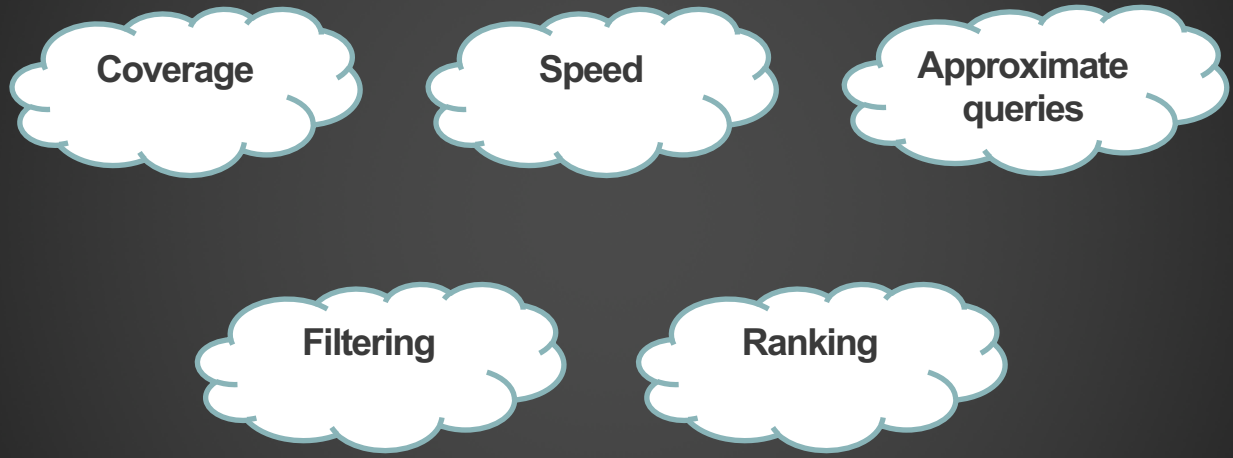
Hound pros/cons

Pros	Cons
✓ Supports regex and substring searches	✗ Response time heavily depends on query
✓ Case-sensitive and case-insensitive matching	✗ Scalability issues
✓ File path and repo filtering	✗ Limited searching options
✓ APIs	✗ Limited monitoring capabilities
✓ Open-source	✗ Limited additional features

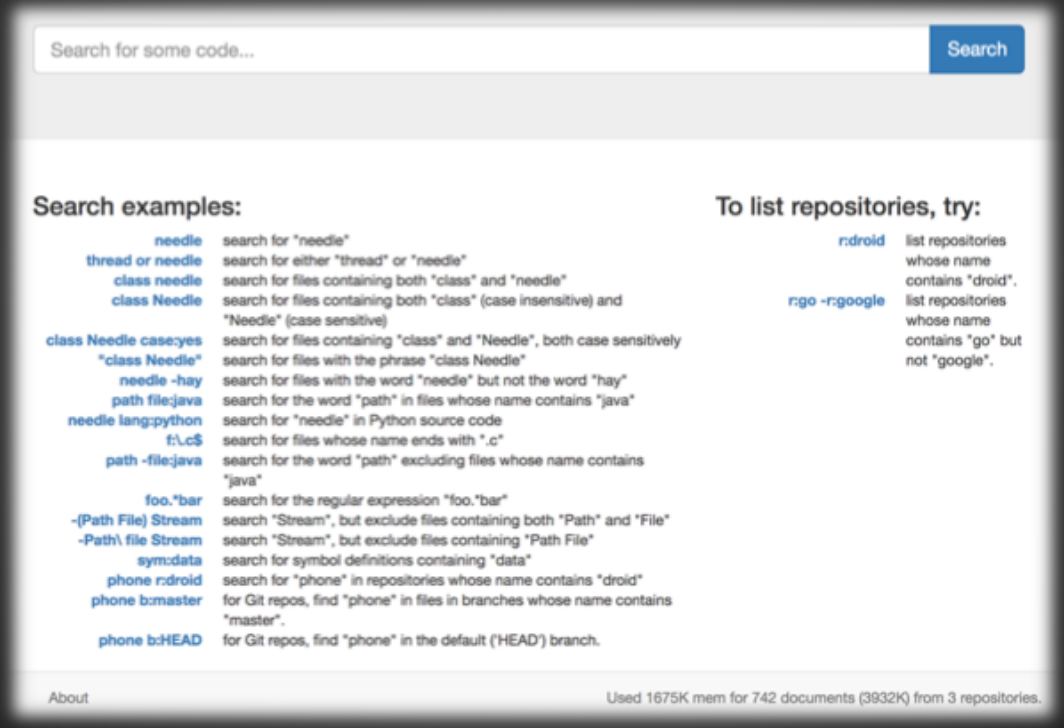


- Zoekt is an open-source fast trigram based source code search engine developed by Google.
- Uses positional ngrams for indexing and matching.
- Developed at Google by Han-Wen Nienhuys.
- 10x faster than Hound, rich support for filtering.

Zoekt design principles



Zoekt screenshot



Zoekt pros/cons

Pros	Cons
✓ Super fast search, consistent speed regardless of search	✗ Poor UI
✓ Sophisticated design and approach for indexing using position trigrams	✗ Limited monitoring
✓ Rich searching options, fully supports regex and substring searches	✗ Limited automation around running and deploying the service
✓ Easy to build features on top of it	✗ Limited additional features
✓ Open-source	✗ No APIs

- Sourcegraph is a fast, solid, full-featured code navigation engine with code intelligence features by Sourcegraph.
- It leverages git grep to find code and uses Zoekt for indexed searches.
- Its language models implement the Language Server Protocol (LSP) to provide Code Intelligence features.
- Developed by Sourcegraph, a company often referred to as the “Google for Code”.

Did you know?

- The open-source Sourcegraph browser extension adds code intelligence to files and diffs on GitHub, GitHub Enterprise, Phabricator, and Bitbucket Server for free!

Sourcegraph demo

Sourcegraph pros/cons

Pros	Cons
✓ Rich features inc. cross-reference and semantic search	✗ Requires many resources to run properly
✓ Excellent support (company dedicated on that)	✗ Constantly changing price
✓ Excellent documentation around setting it up and running it.	✗ Sourcegraph Enterprise price per user
✓ Numerous plugins (e.g. for text editors, IDEs, browsers)	
✓ Core version free	

Comparison

	Searchcode Server	Hound	Zoekt	Sourcegraph
Speed				 *
Scalability				 *
Searching options				
Additional features				
Maintainability				
Support				
License/Price				 **

*Unless deployed to a cluster, the service is relatively slow when compared to its alternatives.
**The basic version of the software is free, but companies would need the Enterprise version for which there's a cost per user. Basic version lacks indexed search and cluster deployments.

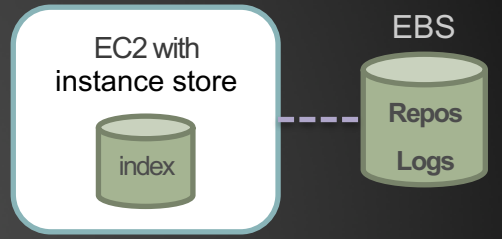
Recommendations

Use:

- 👉 **Searchcode Server**: if you're looking for a more generic search engine, that can be easily maintained and monitored.
- 👉 **Hound**: No reason to use Hound instead of Zoekt.
- 👉 **Zoekt**: if you focus on speed, scalability, and search options.
- 👉 **Sourcegraph**: if you want to invest on a source code search engine and need additional features such as code intelligence and integrations.

Things to consider

- ✍ Use a local SSD (instance store volume) since these services constantly hit the disk.
- ✍ Use permanent storage to store the cloned repos. This allows you to achieve near zero-downtime in case the instance goes down.



- ✍ Make sure you understand and experiment when configuring the services, i.e. in many services you'll need to set limits for max file size, max lines, etc.
- ✍ Monitor logs for errors.
- ✍ Remember, bad documents can always slow down your service!

Recent advances

- ✌ GitHub is experimenting with semantic code search[1,2].
- ✌ Microsoft offers semantic code search for Azure repos in Azure DevOps Services and TFS[3].
- ✌ Google offers fast code search for its Cloud Source Repositories. It's searching options are quite similar to Zoekt's[4]. Plus Google's Bazel code search.
- ✌ Sourcegraph becomes more and more popular by adding more languages to its Code Intelligence feature (thanks to Microsoft's Language Server Protocol) and by providing more integrations and open-source browser extensions.

[1] "Towards Natural Language Semantic Code Search": https://githubengineering.com/towards-natural-language-semantic-code-search/
[2] "GitHub experiments; semantic code search.": https://experiments.github.com/semantic-code-search
[3] "Search across all your code and work items": https://docs.microsoft.com/en-us/azure/devops/project/search/overview?view=vsts&tabs=new-nav
[4] "Searching for code": https://cloud.google.com/source-repositories/docs/searching-code

PART 2

Searching for API usage examples

CLAMS

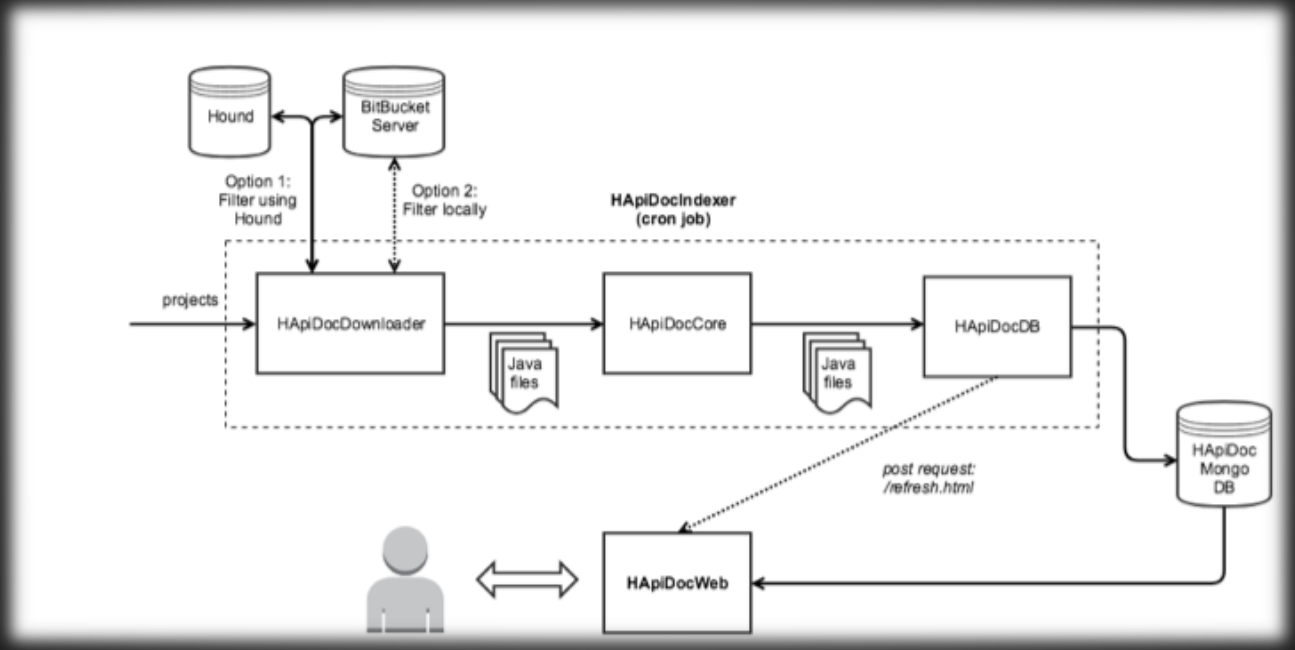
The Problem

- 👎 *Lack of proper documentation for the APIs*
- 👎 *How can I use this API/API method?*
- 👎 *Creating API usage examples is time-consuming*

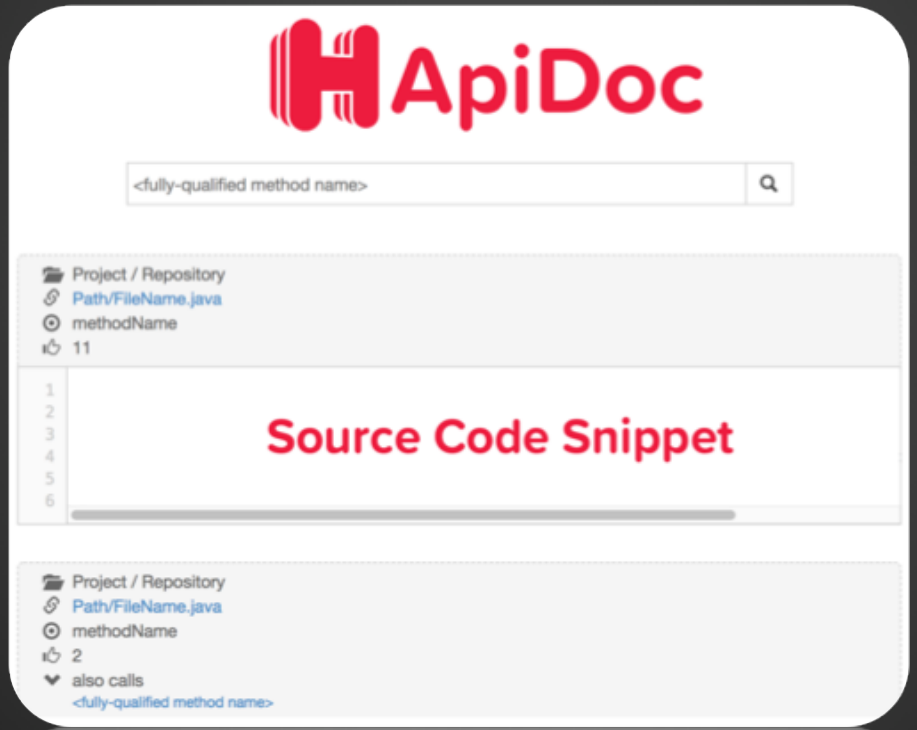
The Concept

- ✌️ *What if we mine examples from client source code?*
- ✌️ *Would be nice to cluster results*
- ✌️ *And show the most indicative example(s) of each cluster*
- ✌️ *And provide a summarised version of the most indicative example(s)*

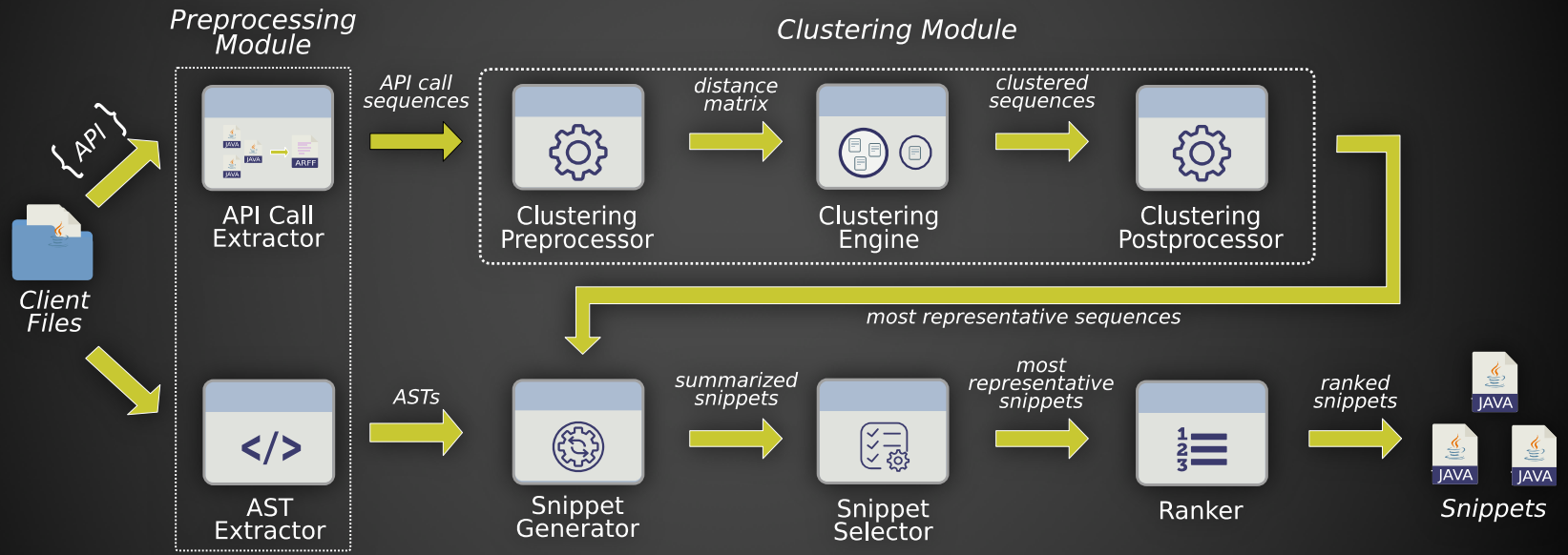
HApiDoc Architecture



HApiDoc UI


HApiDoc
<fully-qualified method name>
Project / Repository
Path/FileName.java
methodName
11
Source Code Snippet
Project / Repository
Path/FileName.java
methodName
2
also calls
<fully-qualified method name>

Behind the scenes - CLAMS



Closing Notes

- ✌ HApiDoc is getting open-sourced! Looking for contributors!
- ✌ Take a look at our GitHub space: https://github.com/HotelsDotCom
- ✌ Presentation material: https://github.com/nikos912000/voxxed-thes-material
- ✌ Using any other code search engines? Let us know!

PS: We're hiring!

Thank you!

 [1]

nikos912000@notmail.com

 [2]

@nikos912000

 [3]

www.linkedin.com/in/nkatirtzis

 [4]

nikos912000

[1] Icon made by Gregor Cresnar from www.flaticon.com.
[2] Icon made by Freepik from www.flaticon.com.
[3] Icon made by Freepik from www.flaticon.com.
[4] Icon made by Pixel Perfect from www.flaticon.com.