

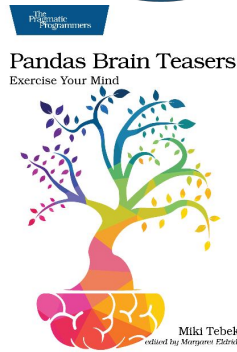
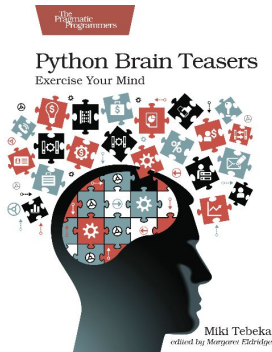
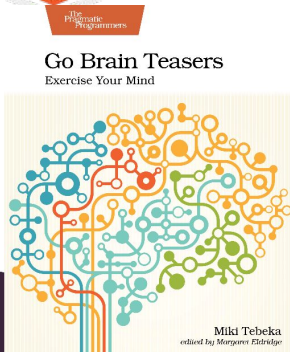
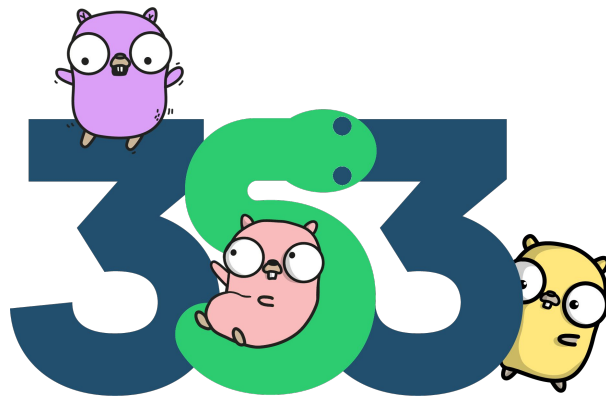
Steaming Data

One Record at a Time

miki @tebeka



CEO,
CTO,
UFO



GopherCON

Israel



info@gophercon.org.il

CFP



SPONSOR



Range Over Function

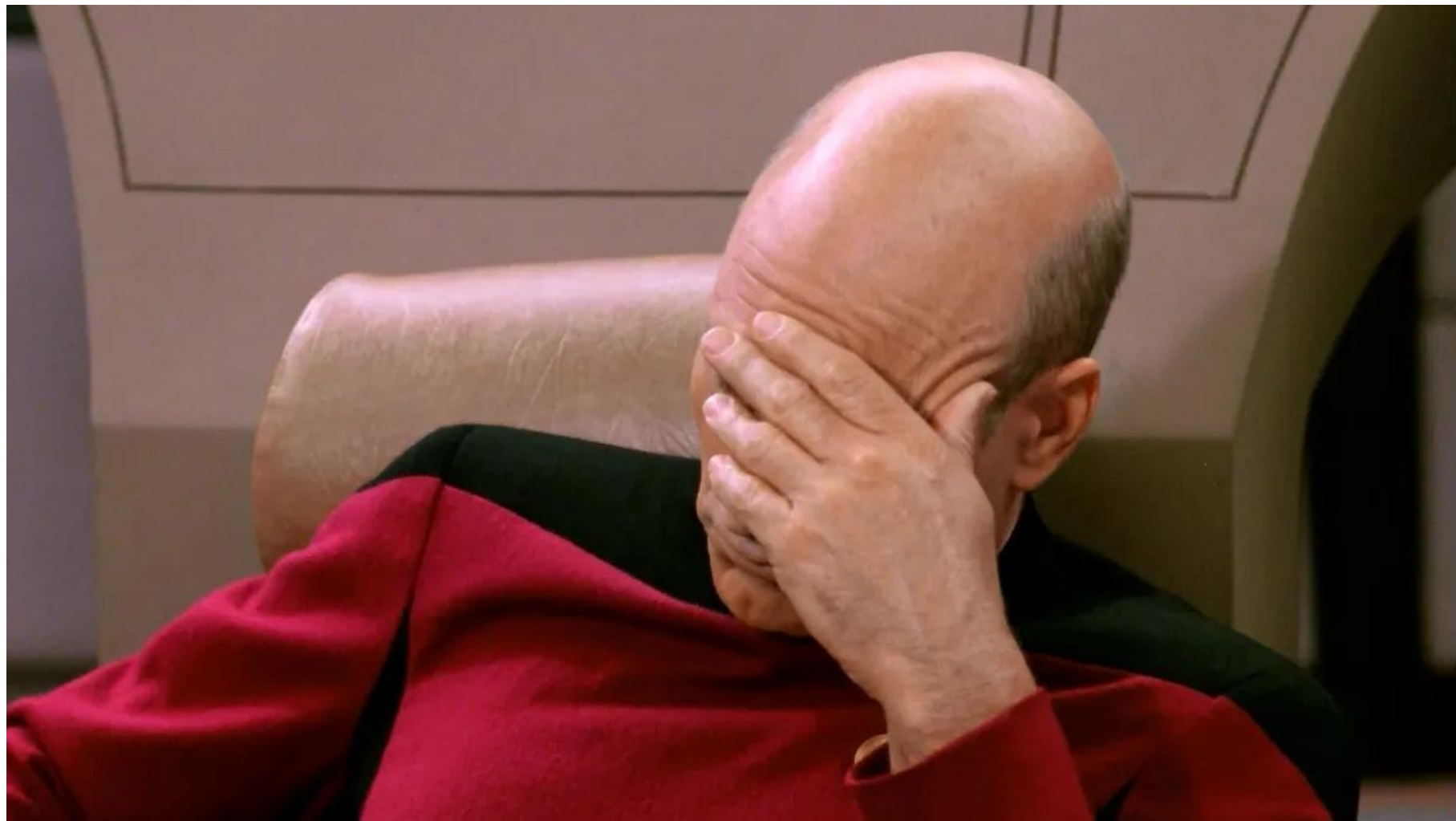
Go 1.23, August 2024

Generators

Python 2.3, July 2003

**Load logs from
file so we can
query them.**

```
$ head logs/1/x00
199.72.81.55 - - [01/Jul/1995:00:00:01
-0400] "GET /history/apollo/ HTTP/1.0" 200
6245
unicomp6.unicomp.net - -
[01/Jul/1995:00:00:06 -0400] "GET
/shuttle/countdown/ HTTP/1.0" 200 3985
199.120.110.21 - - [01/Jul/1995:00:00:09
-0400] "GET
/shuttle/missions/sts-73/mission-sts-73.htm
1 HTTP/1.0" 200 4085
```



Claude Go Regex to Parse Apache Logs ▾

Share



MT

Write a Go regular expression to parse the following logs

```
199.72.81.55 - - [01/Jul/1995:00:00:01 -0400]
"GET /history/apollo/ HTTP/1.0" 200 6245
unicomp6.unicomp.net - - [01/Jul/1995:00:00:06
-0400] "GET /shuttle/countdown/ HTTP/1.0"
200 3985
199.120.110.21 - - [01/Jul/1995:00:00:09 -0400]
"GET /shuttle/missions/sts-73/mission-sts-
73.html HTTP/1.0" 200 4085
burger.letters.com - - [01/Jul/1995:00:00:11
-0400] "GET /shuttle/countdown/liftoff.html
HTTP/1.0" 304 0
199.120.110.21 - - [01/Jul/1995:00:00:11 -0400]
"GET /shuttle/missions/sts-73/sts-73-patch-
small.gif HTTP/1.0" 200 4179
burger.letters.com - - [01/Jul/1995:00:00:12
```

MT

Reply to Claude...



Claude 3.5 Haiku ▾ Choose style ▾

← Apache Log Format Regular Expression Parser ×

```
package main

import (
    "fmt"
    "regexp"
)

// Apache/NCSA Combined Log Format Regex
var logRegex = regexp.MustCompile(`^(\S+) (\S+) (\S+) \[(\S+)\]`

type LogEntry struct {
    Host       string
    Identity   string
    User       string
    Timestamp  string
    Request    string
}
```

Last edited just now



Publish



10 matches (9.86ms) ⓘ



```

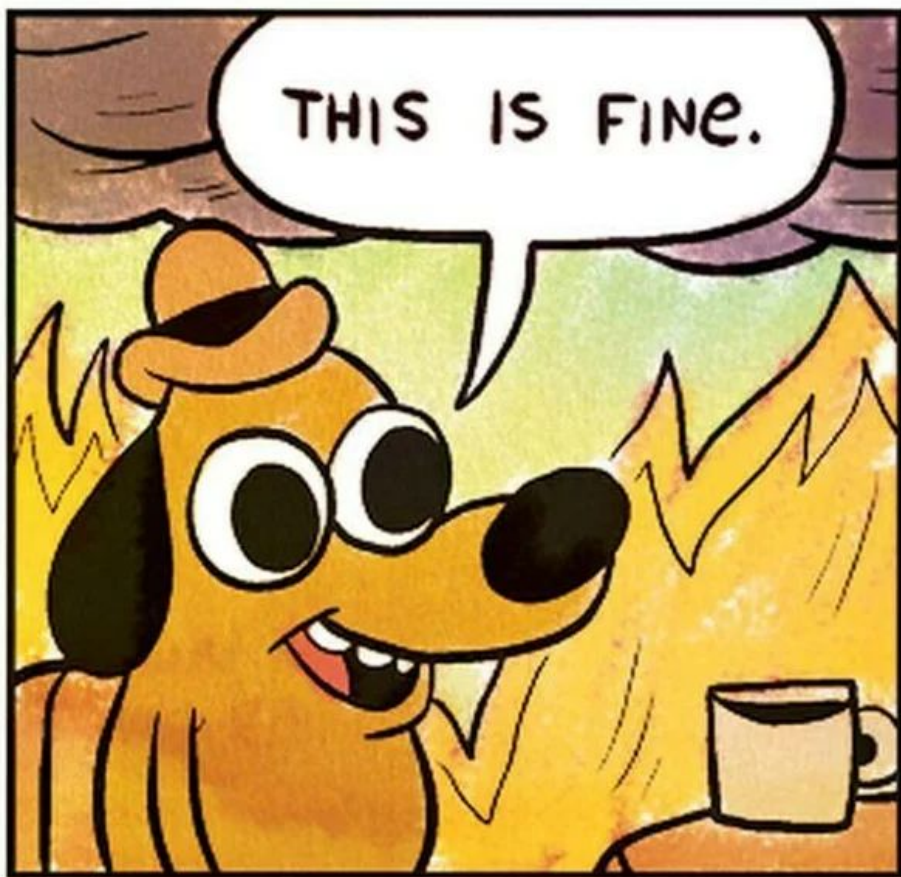
burger.letters.com"GET
/shuttle/countdown/liftoff.html HTTP/1.0"304
199.120.110.21"GET
/shuttle/missions/sts-73/sts-73-patch-small.gif HTTP/1.0"
2004179
burger.letters.com"GET
/images/NASA-logosmall.gif HTTP/1.0"304
burger.letters.com"GET
/shuttle/countdown/video/livevideo.gif HTTP/1.0"200
205.212.115.106"GET
/shuttle/countdown/countdown.html HTTP/1.0"2003985
d104.aa.net"GET
/shuttle/countdown/ HTTP/1.0"2003985
129.94.144.152"GET/
HTTP/1.0"2007074

```

11:1

- asserts position at start of a line
- 1st Capturing Group** `(\S+)`
 - `\S` matches any non-whitespace character (equivalent to `[\r\n\t\f\v]`)
 - matches the previous token between `eng`

| | | |
|---------|---------|-------------------------------|
| Group 1 | 87-107 | unicomp6.unicomp.net |
| Group 2 | 108-109 | - |
| Group 3 | 110-111 | - |
| Group 4 | 113-139 | 01/Jul/1995:00:00:06 -0400 |



CODE

`parser/parser.go`

CODE

loader/simple

CODE

cmd/load

```
$ du -sh logs  
2.2G logs
```


BIG DATA



CODE

loader/seq

Map

lower: ["Suite", "Up"] -> ["suite", "up"]

Filter

≥ 0 : [2, -1, 3, 4] -> [2, 3, 4]

Reduce

sum: [1, 2, 3, 4] \Rightarrow 10

```
SELECT SUM(vol * price)
FROM trades
WHERE symbol = 'IBM';
```

Reduce Map Filter

CODE

loader/mapper

Load logs from
file so we can
query them.

Fast, safe expression language

Common Expression Language (CEL) is an expression language that's fast, portable, and safe to execute in performance-critical applications. CEL is designed to be embedded in an application, with application-specific extensions, and is ideal for extending declarative configurations that your applications might already use.

Use CEL for things like list filters for API calls, validation constraints on protocol buffers, and authorization rules for API requests.

LEARN MORE

GET STARTED

```
// Simple predicates
'tacocat'.startsWith('taco')

// Parameterized predicates over structured data
account.balance >= transaction.withdrawal

// JSON objects
{'sub': '12345678',
 'aud': 'example2.cel.dev',
 'iss': 'https://example1.cel.dev/jwt-issuer'}

// Strongly typed objects
common.GeoPoint{ latitude: 10.0, longitude: -5.5 }
```



Fast

Accelerated expression evaluation in performance-critical paths from nanoseconds to microseconds.



Portable

Developer friendly, light weight with common syntax across multiple Google and external systems.



Extensible

Supports subsetting and extension, easy to embed and tailor to configuration and policy requirements.



Safe

Non-Turing complete, and only accesses data provided by the host application.



Is CEL right for your project?

CEL is ideal for performance-critical applications because it was designed to evaluate safely and quickly (nanoseconds to

`github.com/google/cel-go`

Transfer-Encoding: chunked

jsonlines

ndjson

CODE

`cmd/httpd`



Questions?



github.com/tebeka/talks/tree/master/streams