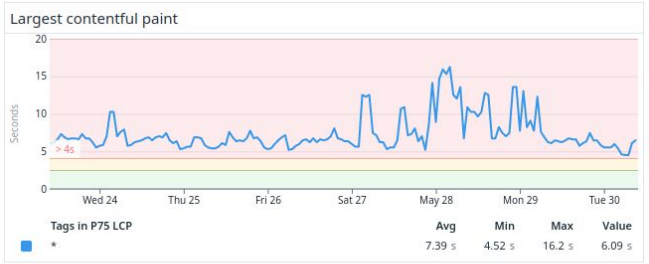
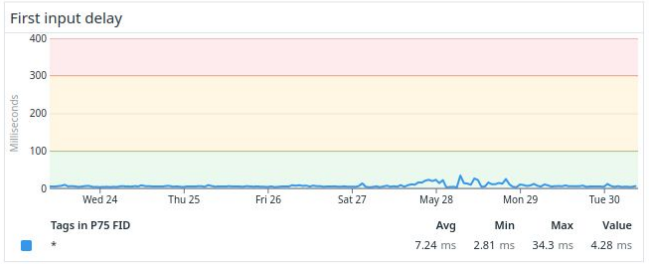
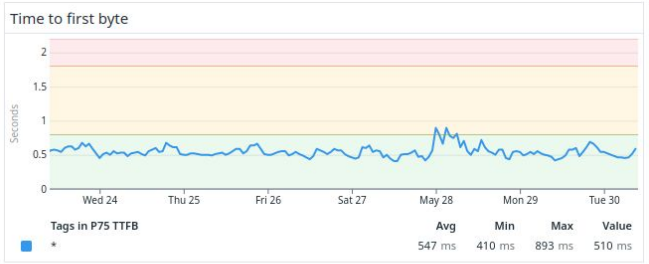
Front-end monitoring

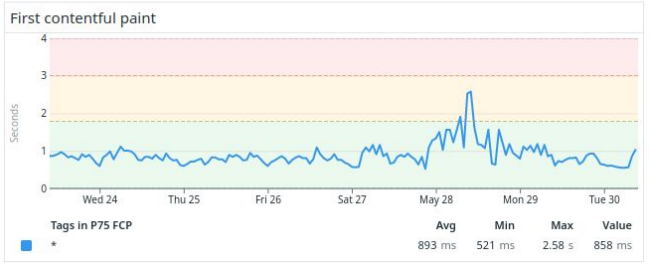
Kryštof Matějka
Site Reliability Engineer

pipedrive









Topics

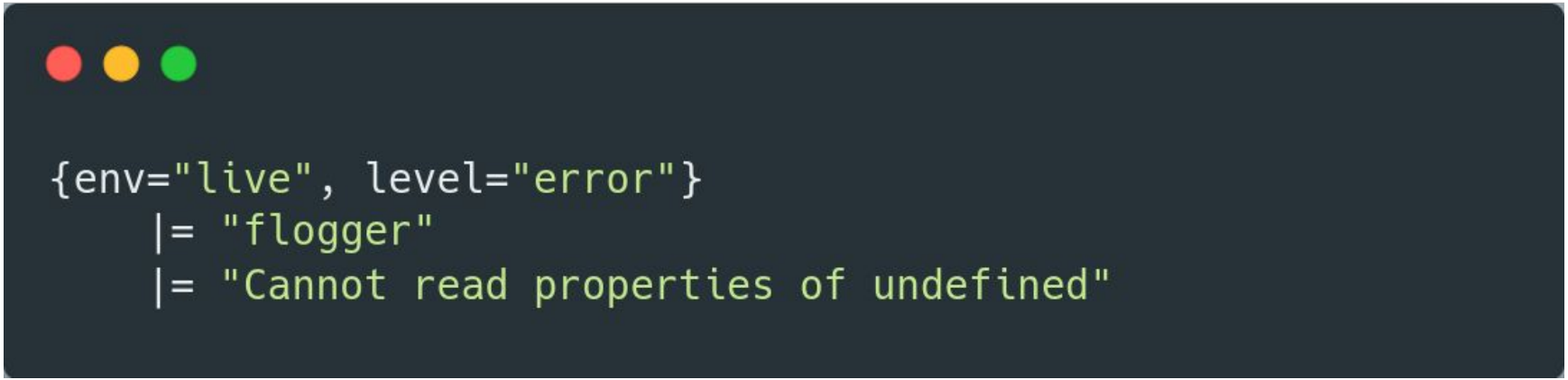
1. Front-end logging
2. Real user monitoring
3. Synthetic monitoring
4. Front-end debugging

Front-end logging

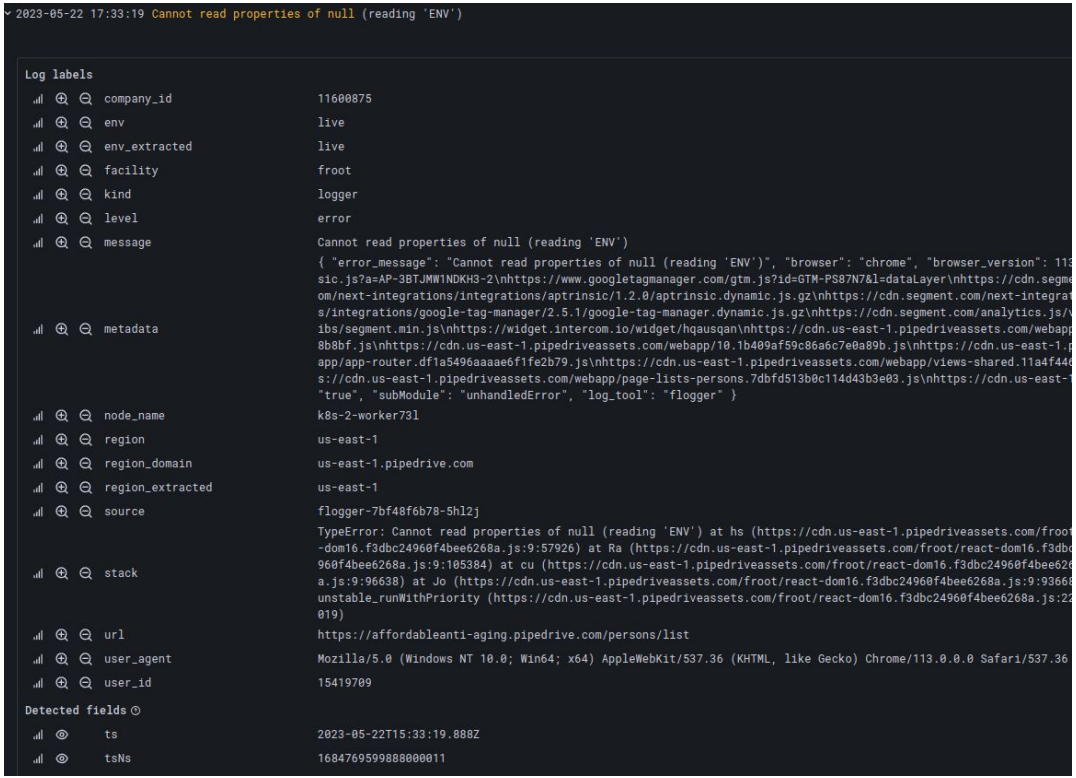
Usage in the browser

```
import logger from '@pipedrive/logger'

logger.info(`this went well`)

logger.error(`something went wrong, ${error}`)
```

LoqQL

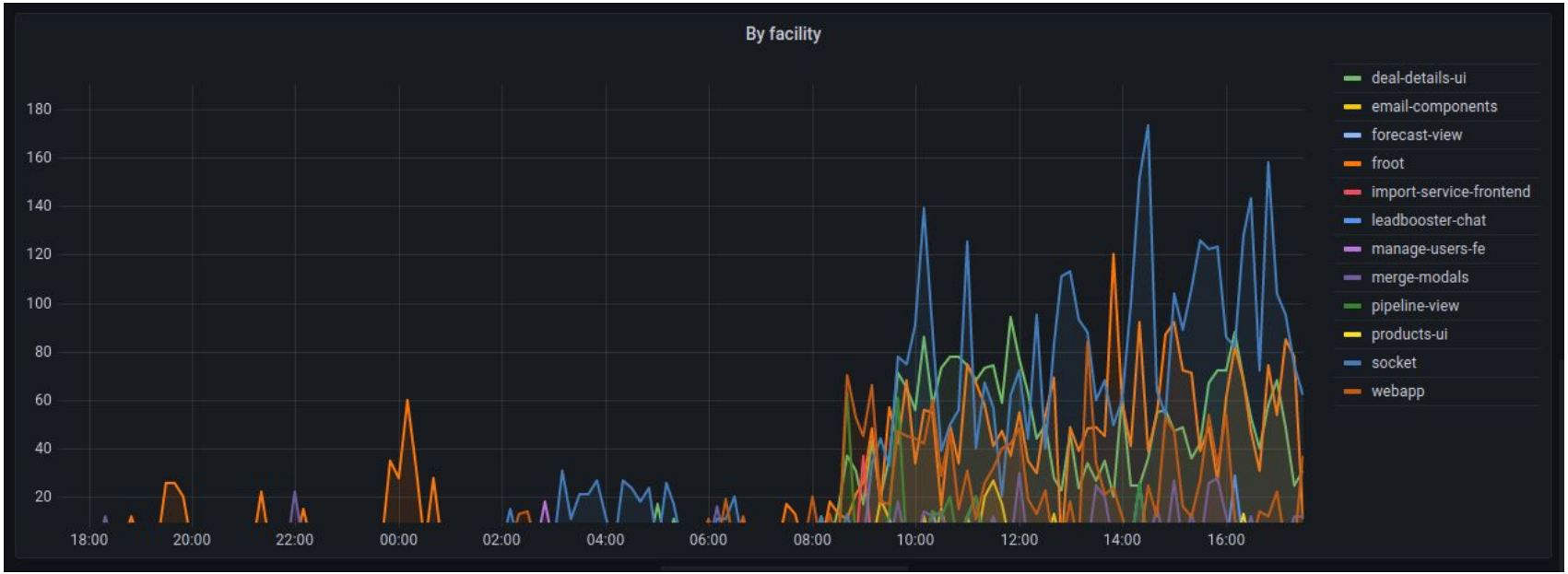

{env="live", level="error"}
|= "flogger"
|= "Cannot read properties of undefined"

Log line example

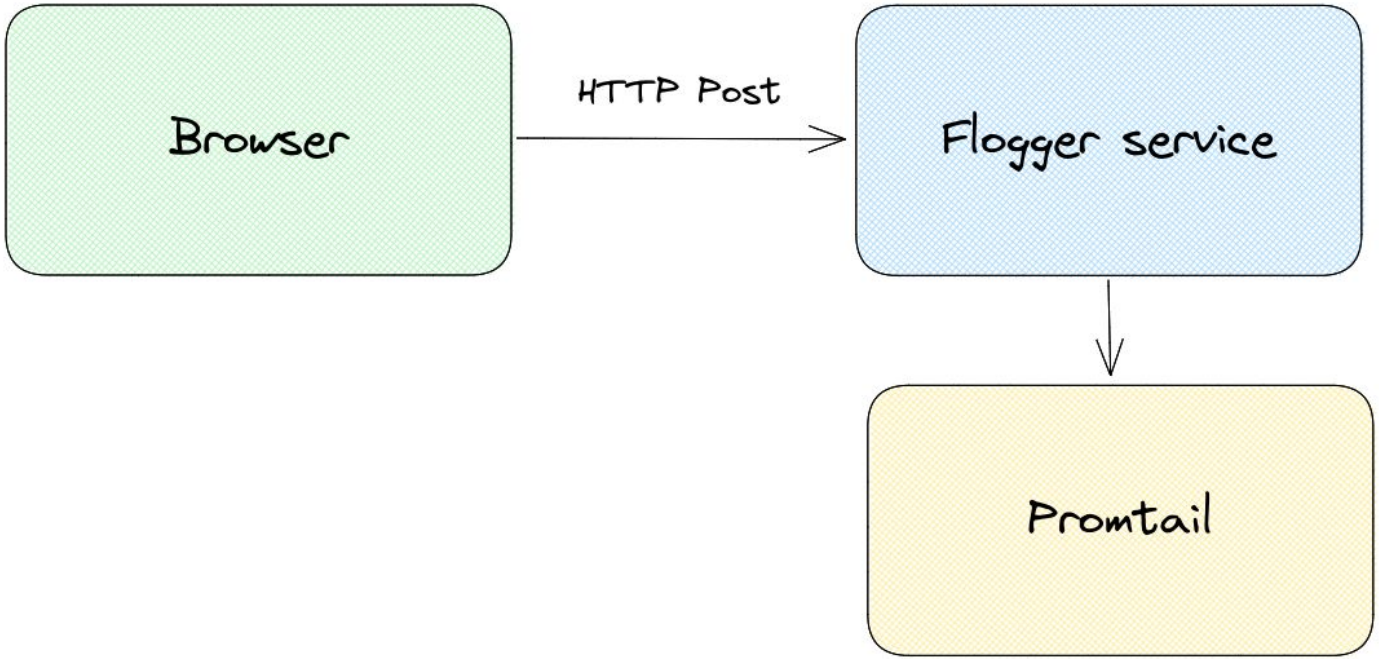


pipedrive

Metric example



Flogger architecture



Be aware

- Front-end logs are not for free
- Be prepared for data loss
- Be careful with rate limiting
- Outage means no logs

Global handler

```
window.addEventListener('error', (event) => {
    //...
})

window.addEventListener('unhandledrejection', (event) => {
    //...
})
```

Anything that can fail

```
try {
    return localStorage.getItem('preferences')
} catch (error) {
    logger.error(`unable to get preferences, ${error}`)
    return null
}
```

When there is a direct impact

```
<ErrorBoundary
    fallback={
        <button onClick={resetBoundary}>Try again!</button>
    }
    onError={logger.error}
>
    <Component/>
</ErrorBoundary>
```

Real user monitoring

Load events

Document: DOMContentLoaded event

When the initial HTML document has been completely loaded and parsed

```
addEventListener("DOMContentLoaded", (event) => {})
```

Window: load event

When the whole page including all dependent resources such as stylesheets, iframes, images and synchronous scripts have been loaded

```
addEventListener("load", (event) => {})
```

Performance API

- Largest contentful paint
- Layout shift
- Element timing
- Event timing
- Long task timing
- Paint timing
- Resource timing
- Attribution timing

Resource timing

- TCP handshake
- DNS lookup time
- Redirection time
- Request time
- TLS negotiation time
- Time to fetch
- ServiceWorker processing time
- Checking if content was compressed
- Checking if local caches were hit
- And more…

Performance API - Resource Timing

```
const observer = new PerformanceObserver((list) => {
  list.getEntries().forEach((entry) => {
    console.log(entry)
  })
})

observer.observe({ type: "resource", buffered: true })
```

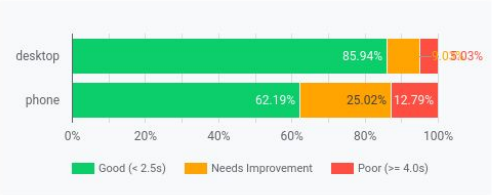
Web vitals

```
import {onLCP} from 'web-vitals'

onLCP(console.log)
```

Chrome UX Report

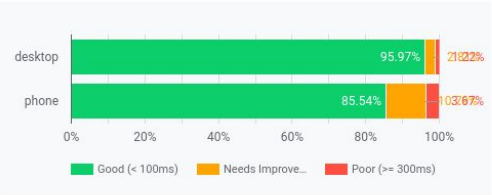
Largest Contentful Paint (LCP)

LCP reports the render time of the largest content element that is visible within the viewport.
web.dev/lcp



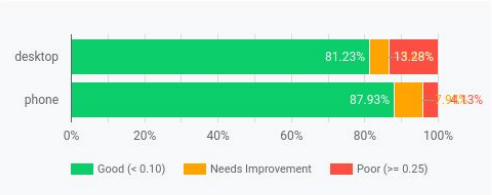
First Input Delay (FID)

FID measures the time from when a user first interacts with a page (i.e. when they click a link, tap on a button, or use a custom, JavaScript-powered control) to the time when the browser is actually able to respond to that interaction.
web.dev/fid



Cumulative Layout Shift (CLS)

CLS measures the sum total of all individual layout shift scores for every unexpected layout shift that occurs during the entire lifespan of the page.
web.dev/cls



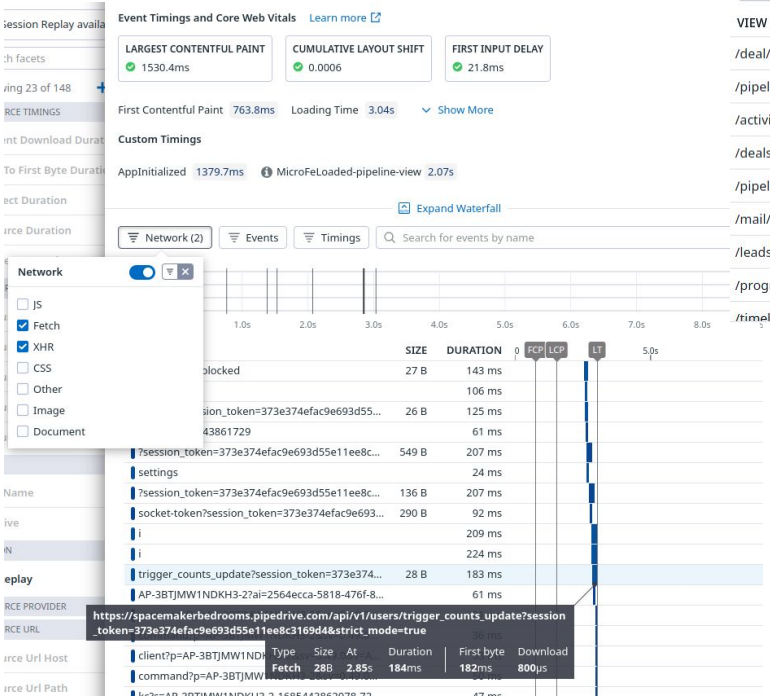
- Data from public websites
- Data collected from Google Chrome
- Api access
- developer.chrome.com/docs/crux

pipedrive

3rd party tools

- Datadog
- Grafana Faro
- New relic
- Sentry
- AppDynamics
- Raygun
- LogicMonitor
- LogRocket
- And more...

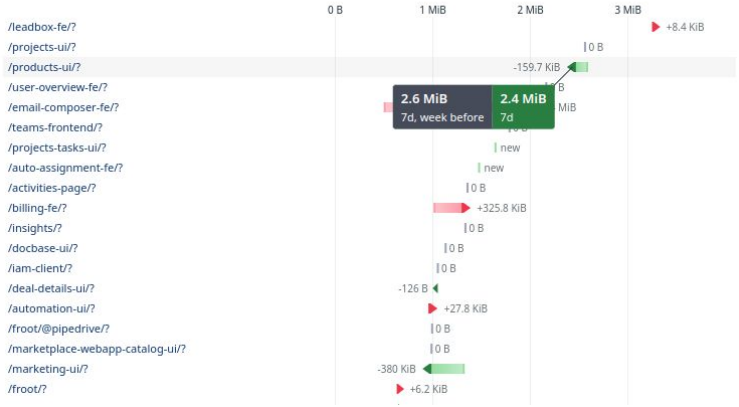
Datadog examples



P75 core web vitals

Search

VIEW NAME	COUNT	LARGEST CONTENTFUL PAINT	FIRST CONTENTFUL PAINT	FIRST INPUT DELAY	FIRST BYTE
/deal/?	325,939	7.15 s	968 ms	6.02 ms	553 ms
/pipeline/?/user/?	107,421	7.90 s	746 ms	4.73 ms	623 ms
/activities/list/user/?	35,090	7.44 s	689 ms	5.45 ms	510 ms
/deals/filter/?	32,230	10.67 s	893 ms	5.03 ms	575 ms
/pipeline/?/user/everyone	30,511	6.60 s	777 ms	—	587 ms
/mail/inbox	27,367	6.34 s	968 ms	10.54 ms	675 ms
/leads/inbox/?	11,990	7.75 s	1,028 ms	10.97 ms	675 ms
/progress/insights/dashboard/?	10,075	7.01 s	649 ms	4.03 ms	500 ms
/timeline/month/?/?	3,184	7.75 s	1,028 ms	10.12 ms	761 ms



pipedrive

Synthetic monitoring

Synthetic monitoring

- Testing and monitoring in controlled environment
- Specific hw, predictable network, specific conditions
- No external interference like extensions or ads
- Also useful in the early stage when you don’t have any users

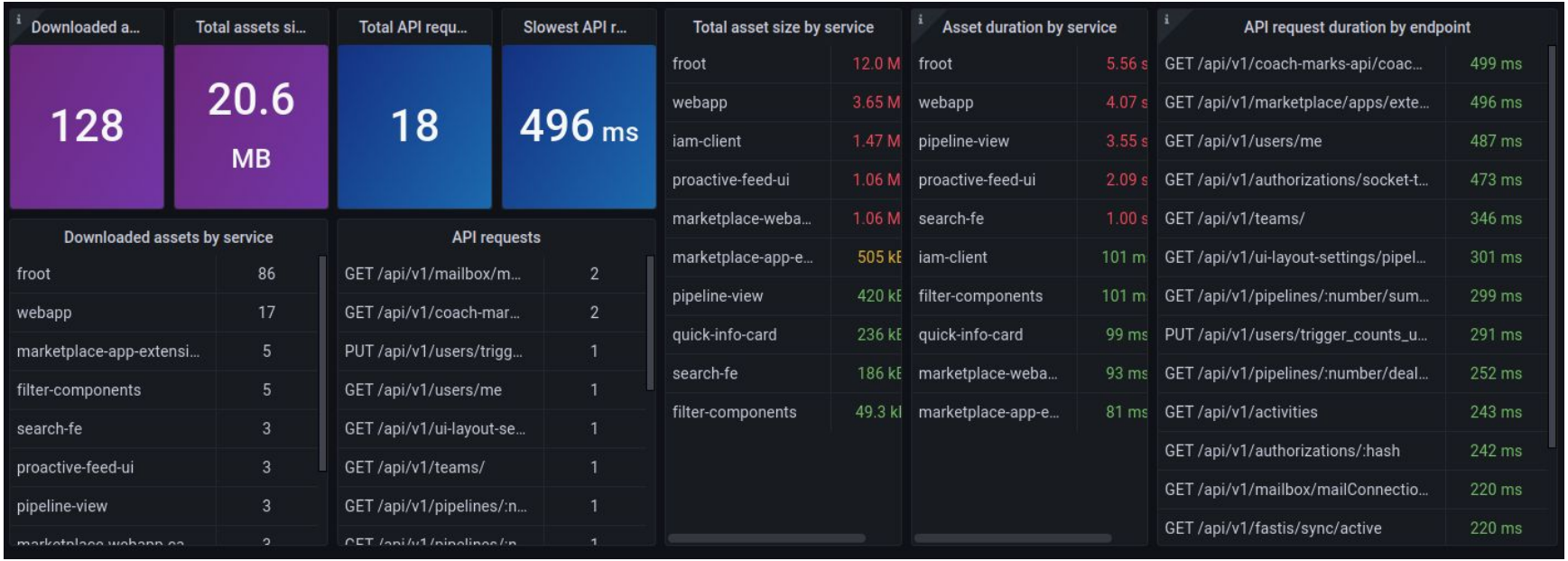
Google lighthouse

```
apt-get install chromium

npm i -g lighthouse

lighthouse --chrome-flags="--headless" --output-path=./report.json --output json --preset=desktop
https://www.pipedrive.com
```

3rd party tools

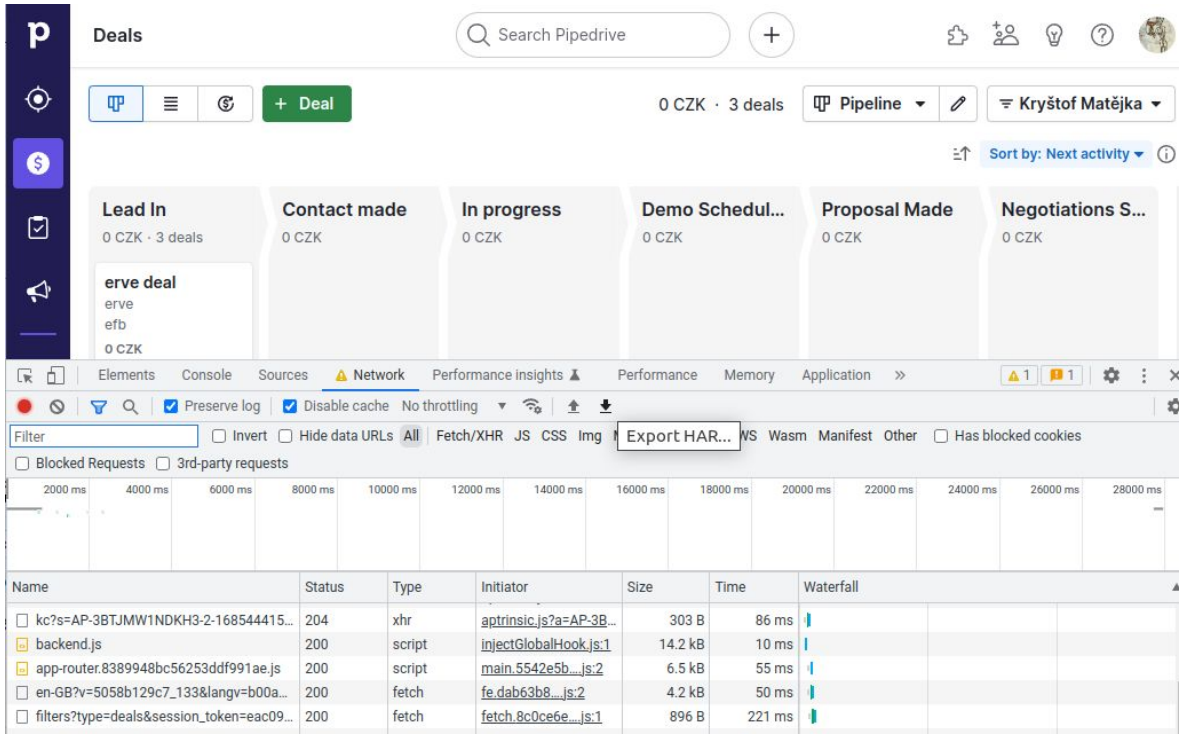
- Datadog synthetic monitoring
- New Relic synthetic monitoring
- And more...

Cypress


Downloaded a...
128
Total assets si...
20.6 MB
Total API requ...
18
Slowest API r...
496 ms
Downloaded assets by service
froot 86
webapp 17
marketplace-app-extensi... 5
filter-components 5
search-fe 3
proactive-feed-ui 3
pipeline-view 3
API requests
GET /api/v1/mailbox/m... 2
GET /api/v1/coach-mar... 2
PUT /api/v1/users/trigg... 1
GET /api/v1/users/me 1
GET /api/v1/ui-layout-se... 1
GET /api/v1/teams/ 1
GET /api/v1/pipelines/:n... 1
Total asset size by service
froot 12.0 M
webapp 3.65 M
iam-client 1.47 M
proactive-feed-ui 1.06 M
marketplace-weba... 1.06 M
marketplace-app-e... 505 kB
pipeline-view 420 kB
quick-info-card 236 kB
search-fe 186 kB
filter-components 49.3 kB
Asset duration by service
froot 5.56 s
webapp 4.07 s
pipeline-view 3.55 s
proactive-feed-ui 2.09 s
search-fe 1.00 s
iam-client 101 ms
filter-components 101 ms
quick-info-card 99 ms
marketplace-weba... 93 ms
marketplace-app-e... 81 ms
API request duration by endpoint
GET /api/v1/coach-marks-api/coac... 499 ms
GET /api/v1/marketplace/apps/exte... 496 ms
GET /api/v1/users/me 487 ms
GET /api/v1/authorizations/socket-t... 473 ms
GET /api/v1/teams/ 346 ms
GET /api/v1/ui-layout-settings/pipel... 301 ms
GET /api/v1/pipelines/:number/sum... 299 ms
PUT /api/v1/users/trigger_counts_u... 291 ms
GET /api/v1/pipelines/:number/deal... 252 ms
GET /api/v1/activities 243 ms
GET /api/v1/authorizations/:hash 242 ms
GET /api/v1/mailbox/mailConnectio... 220 ms
GET /api/v1/fastis/sync/active 220 ms

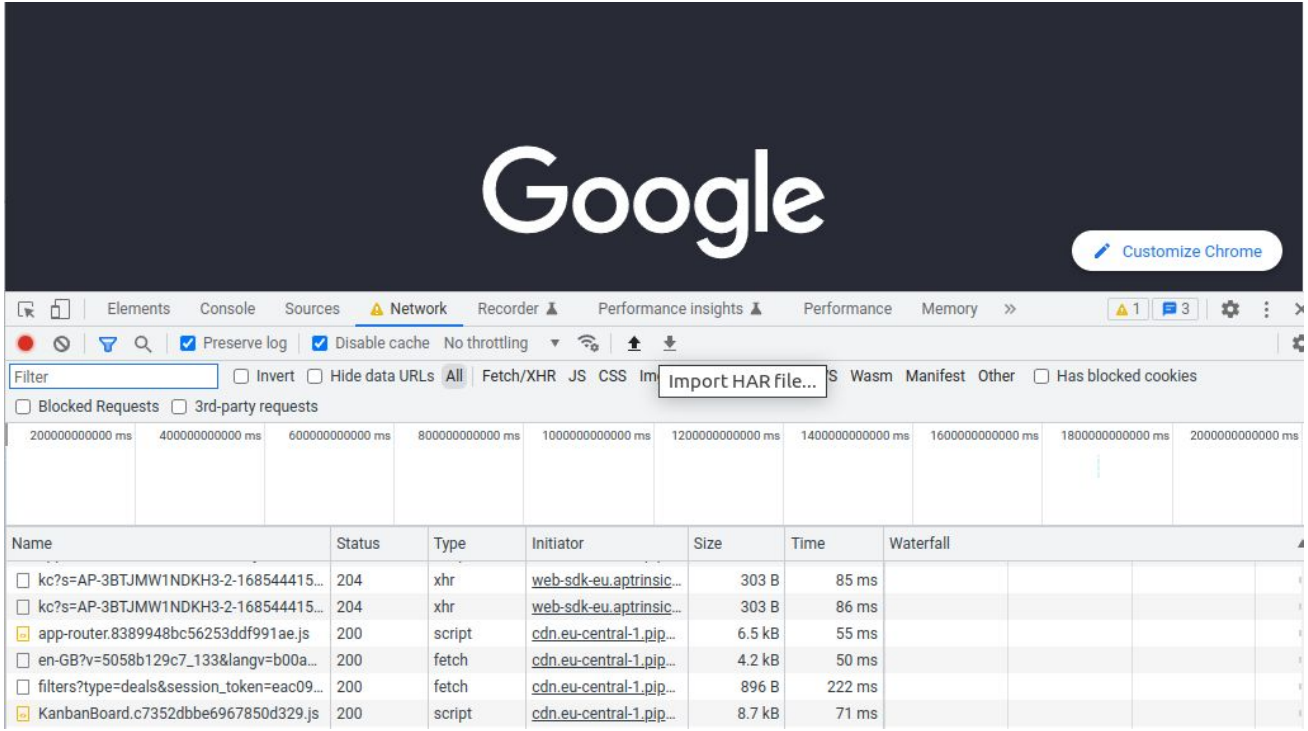
Front-end debugging

HAR - Export



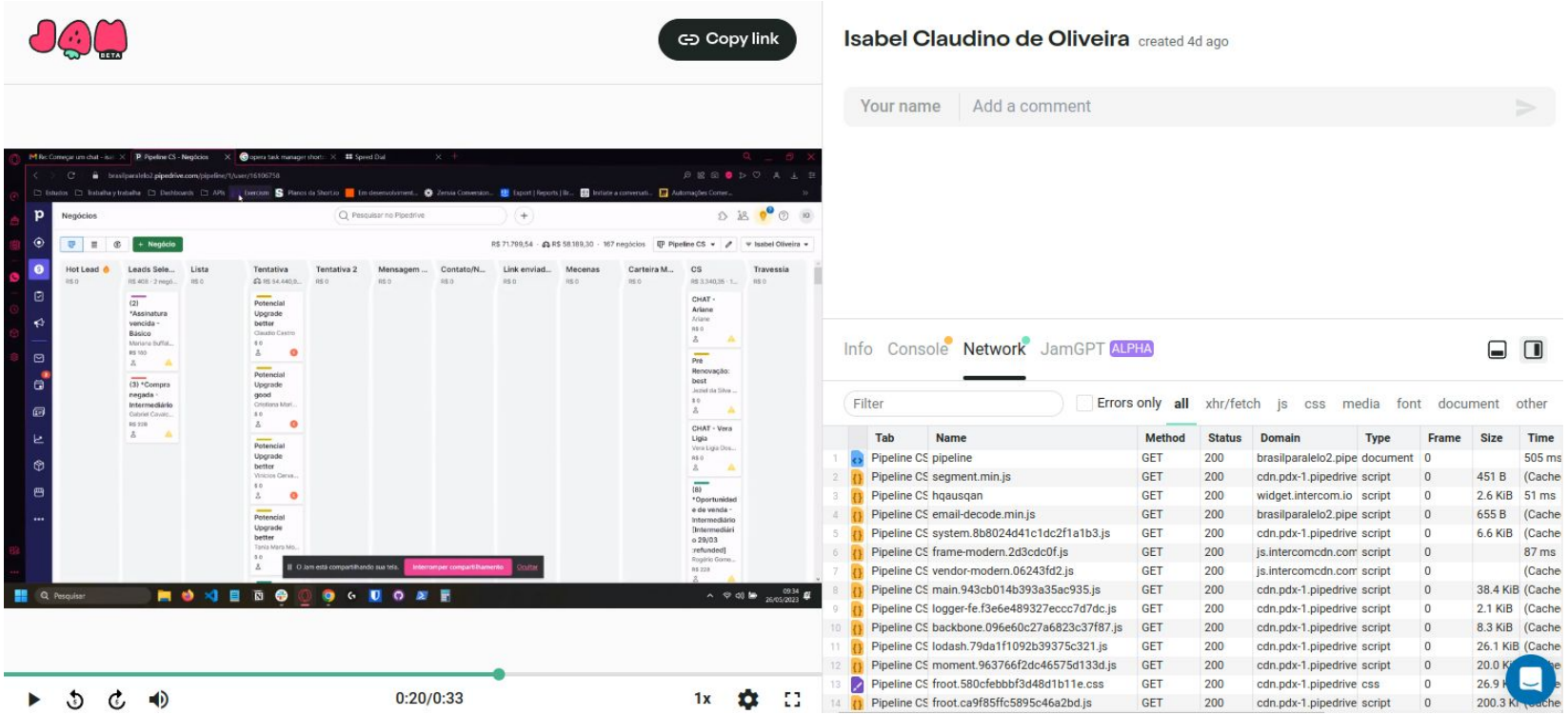
pipedrive

HAR - Import



pipedrive

Jam.dev



pipedrive

Thank you!

Time for your questions