RUM Community Group

2025-05-09 Edition

Agenda

- Logistics
- Next Meeting
- Agenda
  - Header adoption (`Timing-Allow-Origin`, `Server-Timing`)

Logistics

- Meeting cadence
  - 2nd Friday of the Month
  - 60 minutes
  - 10am ET
- Agenda Document
  - (bit.ly/rumcg-agenda)
- Google Meet for Meetings
  - sub to rumcg-participants for invites
- Chat on Web Performance Slack
  `#w3c-rum-community-group`
  - (invites)
- public-rumcg@w3.org
  - (for discussions)

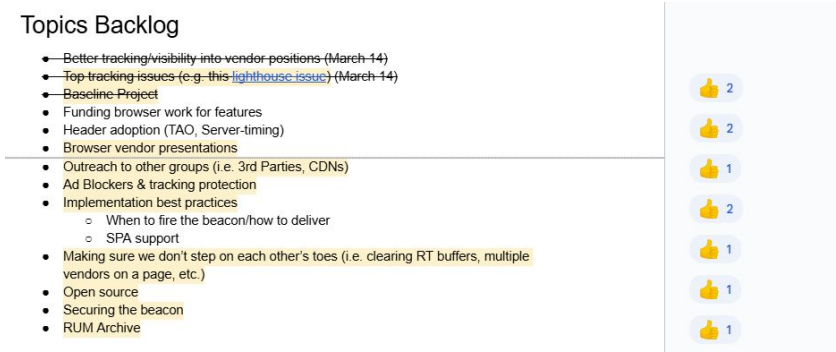
- RUM CG Github Repo
  - github.com/w3c-cg/rum
  - Group details
  - Links
  - Meeting minutes?
- RUM CG Tracking Project
  - github.com/orgs/w3c-cg/projects/1
- Meeting Minutes via fireflies.ai (NEW)
  - AI summary + transcript will be posted after
  - Full video recording also available

Next Meeting

- Fri **June 20th** @ 10am EST
  - ~~2nd~~ (3rd this time!) Friday of each month

- Topics?
  - Barry's Bytes (fetchLater, web-vitals v5?) (Barry Pollard)
  - Firefox `interactionId` (Nazim Can Altinova)
  - Unresponsive crash reports, Self Profiling (Issac Gerges)
  - *(propose your topic here!)*

- Submit & Vote topics @ bit.ly/rumcg-agenda



Header Adoption

- Background
  - `Timing-Allow-Origin`
  - `Server-Timing`
- Benefits for
  - RUM Providers (automatically pick up additional metrics for customers)
  - Individual websites (ST allows peaking into back-end, TAO allows better Page Weight metrics)
- Existing
  - Stats: HTTP Archive, mPulse(?), etc
  - CDN: CloudFlare, Akamai, Fastly, CloudFront, etc
  - Hosting providers: Shopify
  - Individual website
- Goals
  - Increasing adoption
    - CDNS
    - Third-party resources - can highlight best performance / competitive advantage
  - Standardization of Server-Timing
    - Proposals

https://www.w3.org/TR/server-timing/

This specification enables a server to communicate performance metrics about the request-response cycle to the user agent. It also standardizes a JavaScript interface to enable applications to collect, process, and act on these metrics to optimize application delivery.

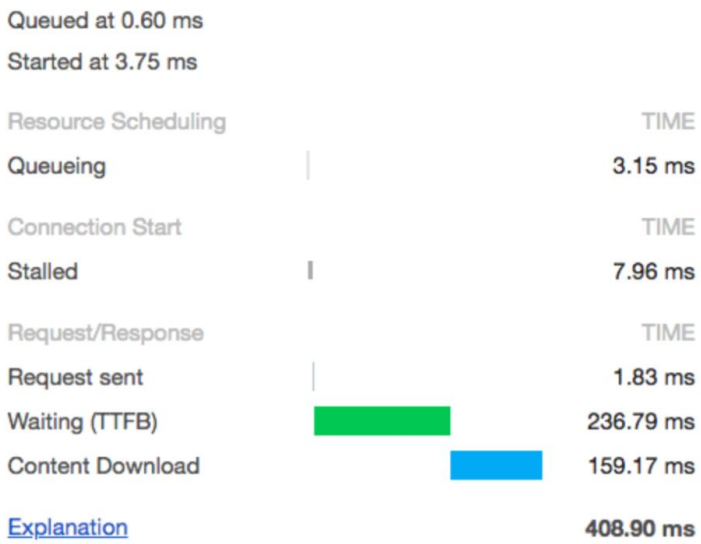
Server Timing

25th Dec 2018 by Charles Vazac

ABOUT THE AUTHOR

Charles Vazac (@vazac) is an engineer working to make the web faster and better at Akamai. As a member of the W3C Web Performance Working Group, he is an editor of the Server-Timing spec. He also was recently selected to the AMP Advisory Committee.

Consider the following timing data for a stylesheet request:



If you were staring at the browser who had to wait around for a response for those 236.79 milliseconds, you'd be hard pressed to find out what was going on. Maybe that time reflects RTT (roundtrip time) and my server responded instantly. Or maybe my server had to do a bunch of custom work to hand me back the bytes of my stylesheet. If I want my TTFB to be shorter, where do I focus my efforts?

CDN use cases

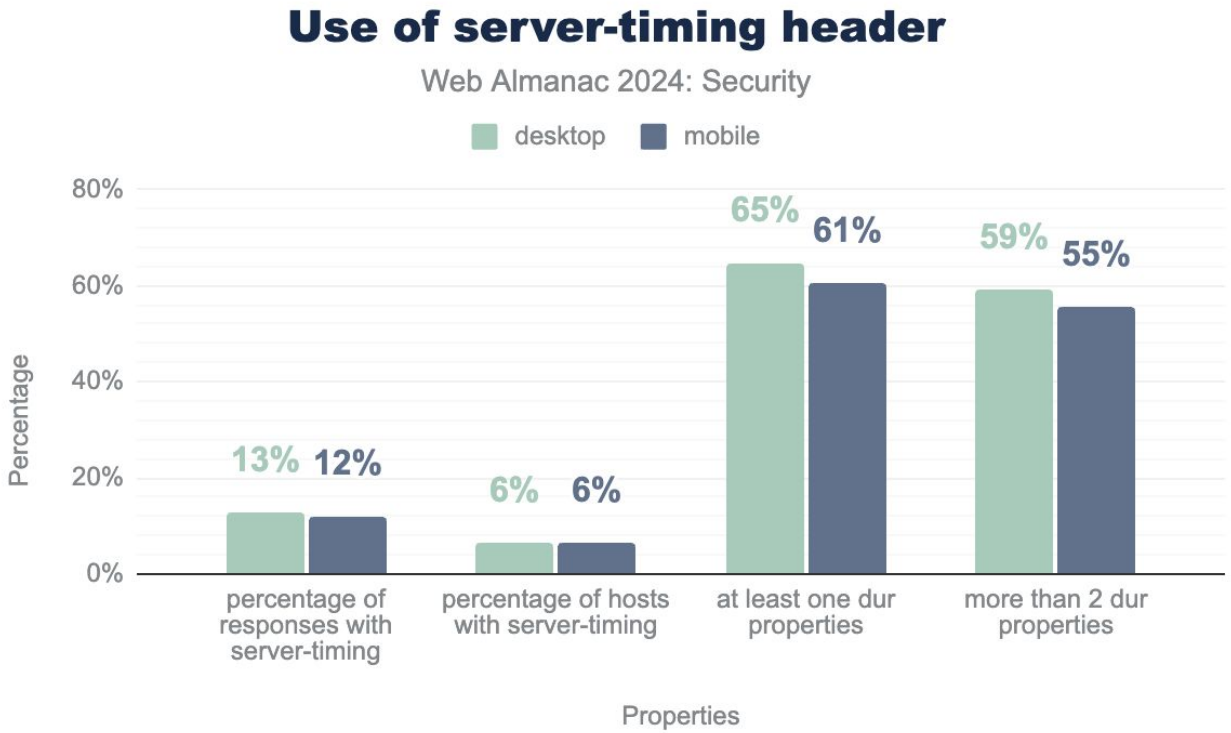
- **Cache Hit/Cache Miss** – Was the resource served from the edge, or did the request have to go to origin?
- **Latency** – How much time does it take to deliver a packet from A to B. Also measured by round trip time (RTT).
- **Origin Time** – How much time did the request spend from your origin? (In the case of a cache miss, this should be zero.)
- **Edge Time** – How much time was spent at the CDN? This can include a lot of different service layers, not just serving from cache. For example, processing of web application firewall (WAF) rules, detecting bots or other malicious traffic though security services, and growing in popularity, edge compute.

Cache/Proxy Info in the Wild

- Akamai (via mPulse)
  - `server-timing: cdn-cache; desc=[HIT|MISS]`
  - `server-timing: edge; dur=[n]`
  - `server-timing: origin; dur=[n]`
- CloudFront (via console option)
  - `server-timing: cdn-upstream-layer;desc="EDGE",cdn-upstream-dns;dur=0,cdn-upstream-connect;dur=69,cdn-upstream-fbl;dur=562,cdn-cache-miss,cdn-pop;desc="DEN52-P3",cdn-rid;desc="5McHcGf1pCMEZKUtTuHH-UI7Co2qq-817CJu_cD7oVUo9BmxBtpIHQ==",cdn-downstream-fbl;dur=563`
- CloudFlare (optional, via workers)
  - `cf-cache-status: [HIT|MISS]`
  - `server-timing: cf_cache;desc=MISC,worker;dur=[n]`
- Fastly (optional, via VCL)
  - `Server-Timing: time-start-msec;dur=1544705663920,time-elapsed;dur=0,fastly-pop;desc=LCY,hit-state;desc=HIT`
- Shopify
  - `server-timing: processing;dur=15, db;dur=5, asn;desc="7922", edge;desc="DFW", country;desc="US", theme;desc="Prestige", pageType;desc="index", servedBy;desc="8jlx", requestID;desc="4ab33c3d-21e6-425a-9754-a6f42a27d36f"`
  - `server-timing: cfRequestDuration;dur=48.999786, earlyhints`

via SpeedCurve blog

Market Adoption - ServerTiming



https://almanac.httparchive.org/en/2024/security

Market Adoption

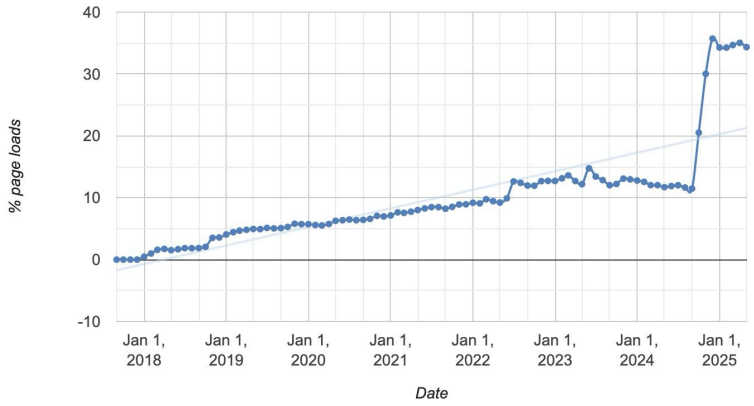
- Chrome Usage: **reading** Server Timing (via RUM) - 34%

← HTML & JavaScript usage metrics > all features > timeline

PerformanceServerTiming Show all historical data: ☑

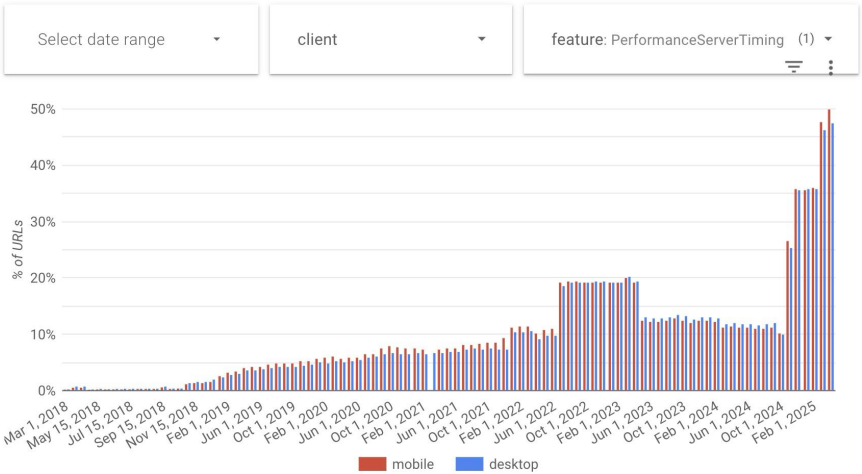
Percentage of page loads over time

The chart below shows the percentage of page loads (in Chrome) that use this feature at least once. Data is across all channels ar platforms. Newly added use counters that are not on Chrome stable yet only have data from the Chrome channels they're on.



Adoption of the feature on top sites

The chart below shows the adoption of the feature by the top URLs on the internet. Data from HTTP Archive.



Common Usage Patterns

n	name	description
7,477,954,019	cdn-cache	HIT,MISS,REVALIDATE,
7,472,902,622	edge	LAX,MIA,IAD,ORD,EWR,SJC,SEA,BNA,,ATL,DFW
1,820,031,999	origin	
143,888,787	cld-fastly	hit,miss
76,662,949	cfExtPri	
30,405,740	cld-akam	hit,hit-near,miss
29,949,816	content-info	
11,719,359	cloudinary	
5,636,156	cache	EDGE,hit-front,PASS
5,214,501	proto	h2,h3,h1
5,202,476	rcomp	ae
5,108,605	pop	TOJ,BRU,NRT,TYO,ITM,FRA,PAR,LCY,MAD,LHR,MUC,SIN,VIE,AMS,HKG,LON,MXP,LIN,ICN,LIS
3,804,892	cfCacheStatus	HIT,DYNAMIC,EXPIRED,MISS
2,643,364	cld-cloudflare	hit,miss
2,607,629	inner	
2,592,911	cfReqDur	
2,167,172	total-response	
1,993,652	cfRequestDuration	
1,963,095	cfHdrFlush	
1,479,460	imagery	
1,479,460	imageryFetch	
920,919	cdntime	
920,919	clientrtt	
920,919	clienttt	
817,759	ef	
817,119	crt	

via 2025 mPulse RUM data (so biased towards Akamai CDN)

Other Use Cases?

Other platform providers?

RUM Support?

- mPulse
- RUM Vision
- SpeedCurve
- ??

What about TAO?

The Timing-Allow-Origin HTTP response header field can be used to communicate a policy indicating origin(s) that may be allowed to see values of attributes that would have been zero due to the cross-origin restrictions.

Market Adoption - TAO

The previous sections showed that the responsibility for third-party negative impact is split between first and third-party developers. However, browsers are also showing interest in optimizing the loading of third-party resources. The proposals include better real user monitoring and developer tooling providing more data about the impact of third parties on their websites.

Figure 8.20. Percent of third-party requests with Timing-Allow-Origin header header.

That might be challenging to achieve given only 25% of total third-party requests provide the Timing-Allow-Origin (TAO) header that is important for third-party web performance data transparency.

Taking into account that the TAO header prevalence has not improved in comparison to the previous years, we would encourage third-party providers to use it more actively, to allow first parties to get more accurate insights into the performance of these resources.

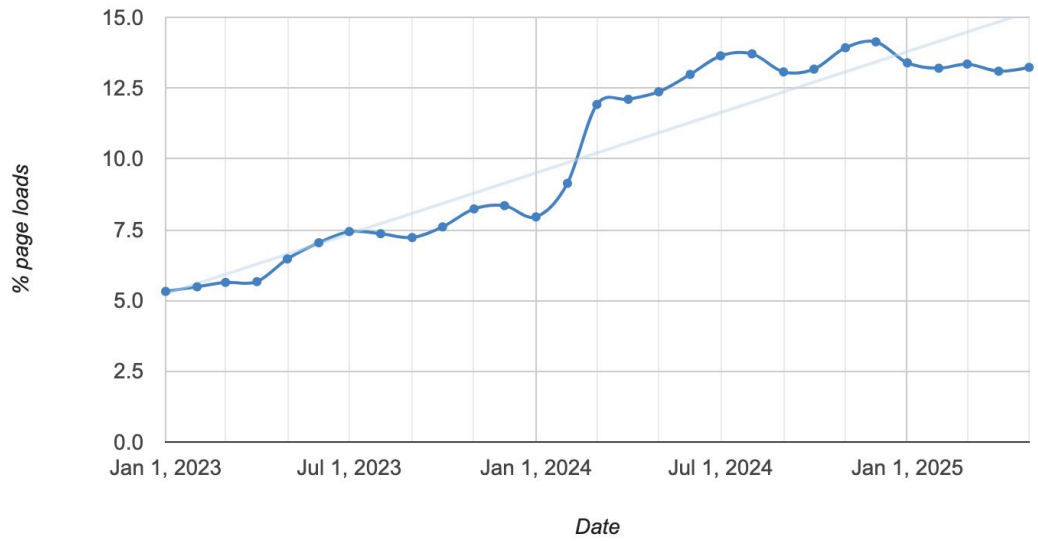
https://almanac.httparchive.org/en/2022/third-parties#fig-20

← HTML & JavaScript usage metrics > all features > timeline

SingleOriginInTimingAllowOrigin Show all historical data: ☐

Percentage of page loads over time

The chart below shows the percentage of page loads (in Chrome) that use this feature at least once. Data is across all channels and platforms. Newly added use counters that are not on Chrome stable yet only have data from the Chrome channels they're on.

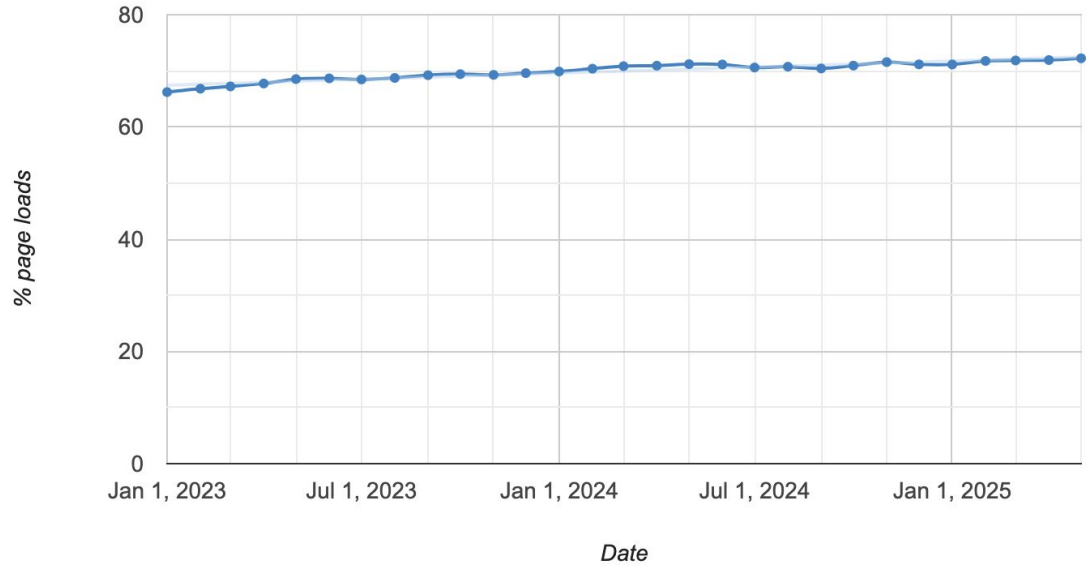


StarInTimingAllowOrigin

Show all historical data: ☐

Percentage of page loads over time

The chart below shows the percentage of page loads (in Chrome) that use this feature at least once. Data is across all channels and platforms. Newly added use counters that are not on Chrome stable yet only have data from the Chrome channels they're on.



Call for Help

- Adoption?
  - Timing-Allow-Origin
  - Server-Timing

- CDNs: Naming conventions?
  - https://www.speedcurve.com/blog/server-timing-time-to-first-byte/
  - e.g. `cdn-cache`; `edge`; `origin`

Ideas

- RUMCG best practices documents
  - TAO for Third-Parties
  - ServerTiming Naming Conventions or Registry
  - Use-Case documents for both of the above

- RUM Archive list of top third-parties not including Timing-Allow-Origin
  - File RUMCG issues for top 3P not using TAO

- HTTP Archive Third-Parties chapter update (in 2022 but not in 2024)

- WebPerfWG anything we can change? IFRAME opt-in?