

---

---

# Shipping a performance API on Chromium

— Experiences from shipping the  
Element Timing API —

---

---

---

---

# Nicolás Peña Moreno

— Google Chrome Speed Metrics —

---

---

# Objectives of talk

- Explain the process involved in standardizing a web performance API and shipping it in Blink.
  - I have a 42 step checklist :)
- Encourage you to get involved!



# Identify a problem

- For performance APIs: gap in measurement.
- Element Timing: measure image render time.

From stevesouders.com blog:

```
  
<script>performance.mark('hero2')</script>
```

This is a hacky solution and does not necessarily provide an accurate timestamp.

# Write an explainer: present problem

Element Timing problem:

- Developers know which are the critical elements.
- Browser knows when content has been painted on the screen.

Shubhie Panicker's initial explainer:

Web developers want to know when the critical elements (aka Hero Elements) on a web page have been displayed on screen.

# Write an explainer: use cases

What user needs can be satisfied?

What are some examples of measurements that would be enabled by the new API?

## Types of Hero Elements

### Hero Elements Displayed on Page Navigation

### Hero Elements Displayed due to User Interaction

# Write an explainer: proposed solution?

A proposed solution is NOT a requirement of an explainer! Not ideal to have a concrete solution.

Element Timing proposal:

- Annotate hero elements
- Expose information via PerformanceObserver

The developer will annotate Hero Elements as follows:  
`<div elementtiming="foobar" class="..." >`

# Socialize explainer

- Present to W3C WebPerf and share explainer.  
<https://lists.w3.org/Archives/Public/public-web-perf/>
- Publish on Web Platform Incubator Group (WICG) Discourse.  
<https://discourse.wicg.io/>



## Element Timing for <img>

TPAC Oct 2018  
npm@chromium.org

# Develop concrete proposal (1)

- Move explainer to WICG on GitHub.  
<https://github.com/WICG/element-timing>
- Request design review from Technical Architecture Group (TAG).

## Element Timing API for images #326

 **Closed** npm1 opened this issue on Nov 16, 2018 · 21 comments

 **npm1** commented on Nov 16, 2018 · edited by kenchriss

こんにちはTAG !

I'm requesting a TAG review of:

- Name: ElementTiming
- Explainer: <https://github.com/WICG/element-timing/blob/master/README.md>
- Spec: <https://wicg.github.io/element-timing/>
- Tests: not yet in WPT, we have [some](#) in Chrome, which we'll move when standardized.
- Primary contacts: [@npm1](#) [@tdresser](#)

# Develop concrete proposal (2)

- Send Intent to Prototype (renamed from Implement).

**Explainer**

**Design doc/Spec**

**Summary**

**Motivation**

**Risks**

**Interoperability and Compatibility**

Edge: [No support](#)  
Firefox: [No support](#)  
Safari: [No support](#)  
Web developers: [Feedback](#) [Examples](#) [Help](#) [More](#)

**Ergonomics**

**Activation**

**Debuggability**

# Multiple Iterations

Updated version of proposal:

<https://docs.google.com/document/d/1sBM5lzDPws2mg1wRKiwM0TGFv9WqI6gEdF7vYhBYqUg/edit>

## OBSOLETE

See [here](#) for more recent iteration of this proposal.

### Element Timing for Images: Explainer

tdresser@, npm@

Created: December, 2017

Last updated March, 2019

**This document is no longer being updated. For the most up to date information, refer to <https://github.com/WICG/element-timing>**

# Implement the proposed API

Chromium Gerrit CHANGES YOUR DOCUMENTATION BROWSE

☆ Merged as 150777e 1197349: [Element Timing] Implement Image Element Timing

Updated Sep 07, 2018

Owner Nicolás Peña Moreno

Uploader Commit Bot

Committer Commit Bot

Assignee Set assignee...

Reviewers Commit Bot  
Chris Harrelson  
Steve Kobes  
ADD REVIEWER

CC SHOW

Repo chromium/src

Branch master

Parent e4aec5d

Topic No topic

Hashtags element-timing X  
ADD HASHTAG

Code-Review +1 Steve Kobes  
+1 Chris Harrelson

Other labels  
Commit-Queue +2 Nicolás Peña Moreno  
Auto-Submit No votes.

REPLY

[Element Timing] Implement Image Element Timing

This CL adds support to observe the first time an <img> element paints via the PerformanceObserver (no support for performance timeline yet) and under a flag that is disabled by default.

Implementation proposal:  
<https://docs.google.com/document/d/1BvHknbj3T-fUuj4RxHF8qGRNxbswtBqUjKiPx7iLXCc>

Intent to Implement:  
<https://groups.google.com/a/chromium.org/forum/#!topic/blink-dev/2twP4Xdd1VI>

Bug: 879270

Cq-Include-Trybots:  
luci.chromium.try:linux\_layout\_tests\_slimming\_paint\_v2;master.tryserver.blink:linux\_trusty\_blink\_rel

Change-Id: I715bd71be088f8055ece493db208caa9b97d736b

Reviewed-on: <https://chromium-review.googlesource.com/1197349>

Commit-Queue: Nicolás Peña Moreno <npm@chromium.org>

Reviewed-by: Chris Harrelson <chrishtr@chromium.org>

Reviewed-by: Steve Kobes <skobes@chromium.org>

Cr-Commit-Position: refs/heads/master@{#589712}

# Add web platform tests: harness

Import the testharness to enable testing:

```
1 <!DOCTYPE HTML>
2 <meta charset=utf-8>
3 <title>Element Timing: observe image inside SVG</title>
4 <script src="/resources/testharness.js"></script>
5 <script src="/resources/testharnessreport.js"></script>
6 <script src="resources/element-timing-helpers.js"></script>
```

# Add web platform tests: image

Remove body margin and insert the hero image:

```
29 <style>
30   body {
31     margin: 0;
32   }
33 </style>
34 <svg width="300" height="300">
35   <image href='resources/circle.svg' elementtiming='my_svg' id='svg_id' />
36 </svg>
```

# Add web platform tests: script

```
13  const observer = new PerformanceObserver(  
14    t.step_func_done(function(entryList) {  
15      assert_equals(entryList.getEntries().length, 1);  
16      const entry = entryList.getEntries()[0];  
17      const pathname = window.location.origin + '/element-timing/resources/circle.svg';  
18      checkElement(entry, pathname, 'my_svg', 'svg_id', beforeRender,  
19        document.getElementById('svg_id'));  
20      // Assume viewport has size at least 200, so the element is fully visible.  
21      checkRect(entry, [0, 200, 0, 200]);  
22      checkNaturalSize(entry, 200, 200);  
23    })  
24  );  
25  observer.observe({entryTypes: ['element']});
```

# Draft spec



## Element Timing API

Draft Community Group Report, 18 November 2019

**This version:**  
<https://wicg.github.io/element-timing>

**Test Suite:**  
<https://github.com/web-platform-tests/wpt/tree/master/element-timing>

**Issue Tracking:**  
GitHub

**Editors:**  
[Nicolás Peña Moreno \(Google\)](#)  
[Tim Dresser \(Google\)](#)

Copyright © 2019 the Contributors to the Element Timing API Specification, published by the Web Platform Incubator Community Group under the W3C Community Contributor License Agreement (CLA). A human-readable [summary](#) is available.

---

### Abstract

This document defines an API that enables monitoring when large or developer-specified image elements and text nodes are displayed on screen.

### Status of this document

This specification was published by the [Web Platform Incubator Community Group](#). It is not a W3C Standard nor is it on the W3C Standards Track. Please note that under the [W3C Community Contributor License Agreement \(CLA\)](#) there is a limited opt-out and other conditions apply. Learn more about [W3C Community and Business Groups](#).

## § 1. Introduction

*This section is non-normative.*

Knowing when critical elements are displayed on screen is key to understanding page load performance. While fast rendering of the essential components is not sufficient for a satisfactory loading experience, it is necessary. Therefore, monitoring these rendering timestamps is important to improve and prevent regressions in page loads.

TABLE OF CONTENTS

- 1 Introduction**
  - 1.1 Elements exposed
  - 1.2 Usage example
- 2 Element Timing**
  - 2.1 PerformanceElementTiming interface
- 3 Processing model**
  - 3.1 Modifications to the DOM specification
  - 3.2 Modifications to the CSS specification
  - 3.3 Modifications to the HTML specification
  - 3.4 Process image that finished loading
  - 3.5 Report image Element Timing
  - 3.6 Report text Element Timing
  - 3.7 Element Timing processing
  - 3.8 Get an element algorithm
- 4 Security & privacy considerations**

**Conformance**

**Index**  
Terms defined by this specification  
Terms defined by reference

**References**  
Normative References  
Informative References

**IDL Index**

Can reach out to experienced spec writer on IRC to get help through this process.

Spec characteristics

- Prose and algorithms
- Written in Bikeshed/ReSpec
- Interactions with other specs (HTML, DOM)
- No Chrome-specific jargon (need to make sense for any implementer).

# Internal launch review

- Performance APIs generally require internal privacy and security reviews.
- WebPerf WG or TAG may also surface privacy and security concerns, and these should be addressed before launching an API.

## Issue 883483: Element Timing for Images: Launch Bug

Reported by [npm@chromium.org](mailto:npm@chromium.org) on Wed, Sep 12, 2018, 5:04 PM EDT

Project Member

 Only users with Google permission can view this issue.

**Feature description:** A web performance API that exposes the first time certain image elements are rendered on the screen.

**Eng owner:** [npm@chromium.org](mailto:npm@chromium.org), [tdresser@chromium.org](mailto:tdresser@chromium.org)

**Product owner:** [npm@chromium.org](mailto:npm@chromium.org), [tdresser@chromium.org](mailto:tdresser@chromium.org)

# (Optional) Origin Trial

<https://github.com/GoogleChrome/OriginTrials/>

- Allows experimenting with a new (not yet shipped) web platform feature!
  - Browser engineers love early feedback.
  - Changes to features after they have shipped can be hard.
- Interested web developers sign up for tokens.
- Only a small portion of page loads can access origin trial.
  - Prevents developers from depending on the experimental feature.

# (Optional) Origin Trial: Intent to Experiment

blink-dev ›

## Intent to Experiment: ElementTiming for Images

3 posts by 2 authors ▾



me (Nicolás Peña [change](#))



**Other recipients:** [tdre...@chromium.org](mailto:tdre...@chromium.org)

### Contact emails

[npm@chromium.org](mailto:npm@chromium.org), [tdresser@chromium.org](mailto:tdresser@chromium.org)

### Spec

Explainer <https://github.com/npm1/Element-Timing>

More detailed doc: <https://docs.google.com/document/d/1blFeMVdqxBOV3BAJh60ptOBFY7cJSXnf7VyW3wspbZ8/>

# (Optional) Origin Trial: feedback

Peter Hedenskog (Wikimedia):

That looks really promising and very accurate, particularly compared to old workarounds. We tested a couple more URLs that you can see in `T219231` and they showed the same result.

For our content, it looks like the Element Timing API finally provides a way for us to know accurately when images are really displayed to users!

We'd love to get more feedback from more developers, but we understand it's a big time commitment to try out an API which may never ship.

# Polish proposal

- Obtain signals from web developers and other browsers.
- WICG spec
- Chromium implementation
- Address feedback from TAG review.

## WebPerf WG agenda

### Interested in participating?

- Jump on [our video chat / hangout](#)
- Find us on IRC: <http://irc.w3.org> #webperf
- Join the mailing list: [public-web-perf@](mailto:public-web-perf@)
- Want to get a calendar invite? Add your name to the [list](#).
- Our meetings are recorded! [Video playlist](#)
- [Dashboard](#)

### Resources

- [Web Performance Working Group charter](#)
- [Spec status spreadsheet](#)
- [WebPerf WPT tests](#)

- [Add fully active check](#) ✓ 3  
#24 by npm1 was merged on Jul 24, 2019 • Approved
- [Remove responseEnd section](#) ✓ 1  
#23 by npm1 was merged on Jul 24, 2019 • Approved
- [Set identifier and id on entry creation](#) ✓ 3  
#22 by npm1 was merged on Jul 24, 2019 • Approved
- [Define `elementTiming` as reflecting "elementtiming"](#) ✓ 2  
#19 by foolip was merged on Jul 24, 2019 • Approved
- [Add loadTime to potentially-add-LCP](#) ✓ 1  
#17 by npm1 was merged on Jul 25, 2019 • Approved
- [Update spec to use loadTime](#) ✓ 13  
#16 by npm1 was merged on Jul 23, 2019 • Approved
- [Set rendering timestamp in renderTime](#) ✓ 1  
#15 by npm1 was merged on Jul 16, 2019 • Approved
- [Add support for data URL images](#) ✓ 17  
#14 by npm1 was merged on Jul 12, 2019 • Approved
- [Add hooks for LCP integration](#) ✓ 13  
#13 by yoavweiss was merged on Jun 25, 2019 • Approved
- [Remove implicit registration from spec](#) ✓ 0  
#10 by npm1 was merged on May 31, 2019
- [Add text support](#) ✓ 27  
#7 by npm1 was merged on May 30, 2019
- [Update image attributes](#) ✓ 0  
#6 by npm1 was merged on May 28, 2019
- [Address feedback](#) ✓ 6  
#5 by npm1 was merged on May 24, 2019

Subject	Status	Owner
☆ Add buffered flag tests for UserTiming and ElementTiming	Merged	 Nicolás Peña Moreno
☆ [ElementTiming] Check opacity of image before exposing	Merged	 Nicolás Peña Moreno
☆ [ElementTiming] Add tests for generated and shadow text	Merged	 Nicolás Peña Moreno
☆ [ElementTiming] Add intersectionRect for text entries	Merged	 Nicolás Peña Moreno
☆ [ElementTiming] Do not walk objects that cannot be reported	Merged	 Nicolás Peña Moreno
☆ [ElementTiming] Text support part 2	Merged	 Nicolás Peña Moreno
☆ [ElementTiming] Change  name  and add  url  to PerformanceElementTiming	Merged	 Nicolás Peña Moreno
☆ [ElementTiming] Change  name  and add  url  to PerformanceElementTiming	Abandoned, WIP	 Nicolás Peña Moreno
☆ [ElementTiming] Add text support part 1	Merged	 Nicolás Peña Moreno
☆ [ElementTiming] Do not expose elements in Shadow Trees	Merged	 Nicolás Peña Moreno
☆ [ElementTiming] Add a test for invisible images	Merged	 Nicolás Peña Moreno
☆ [ElementTiming] Add element attribute	Merged	 Nicolás Peña Moreno
☆ [ElementTiming] Add test for image not added to DOM	Merged	 Nicolás Peña Moreno
☆ [ElementTiming] Fix DCHECK about natural width and height	Merged	 Nicolás Peña Moreno
☆ [ElementTiming] Add shadow image test	Merged	 Nicolás Peña Moreno
☆ [ElementTiming] Improve inline image behavior	Merged	 Nicolás Peña Moreno
☆ [ElementTiming] Add background image support	Merged	 Nicolás Peña Moreno
☆ [ElementTiming] Add element ID	Merged	 Nicolás Peña Moreno
☆ [ElementTiming] Add intrinsic size	Merged	 Nicolás Peña Moreno
☆ [ElementTiming] Mimic FCP++ rect computations	Merged	 Nicolás Peña Moreno
☆ [ElementTiming] Add carousel image test	Merged	 Nicolás Peña Moreno
☆ [ElementTiming] Expose cross origin elements that do not pass TAO	Merged	 Nicolás Peña Moreno
☆ [ElementTiming] Allow timing for video poster image	Merged	 Nicolás Peña Moreno
☆ [ElementTiming] Expose cross origin elements that do not pass TAO	Merged	 Nicolás Peña Moreno
☆ [ElementTiming] Add a test to prevent leak in ImageElementTiming	Merged	 Nicolás Peña Moreno

# Ship new API

- Send Intent to Ship and get approval from 3 Blink API owners.
- Ensure chromestatus.com has accurate information about the API.
- Flip implementation flag to 'enable by default'.

★ **Chris Harrelson** LGTM1

---

★ **Daniel Bratell** LGTM2 /Daniel

---

 **Alex Russell**

★ LGTM3 *conditional on the publication of Origin-Trial survey feedback.*

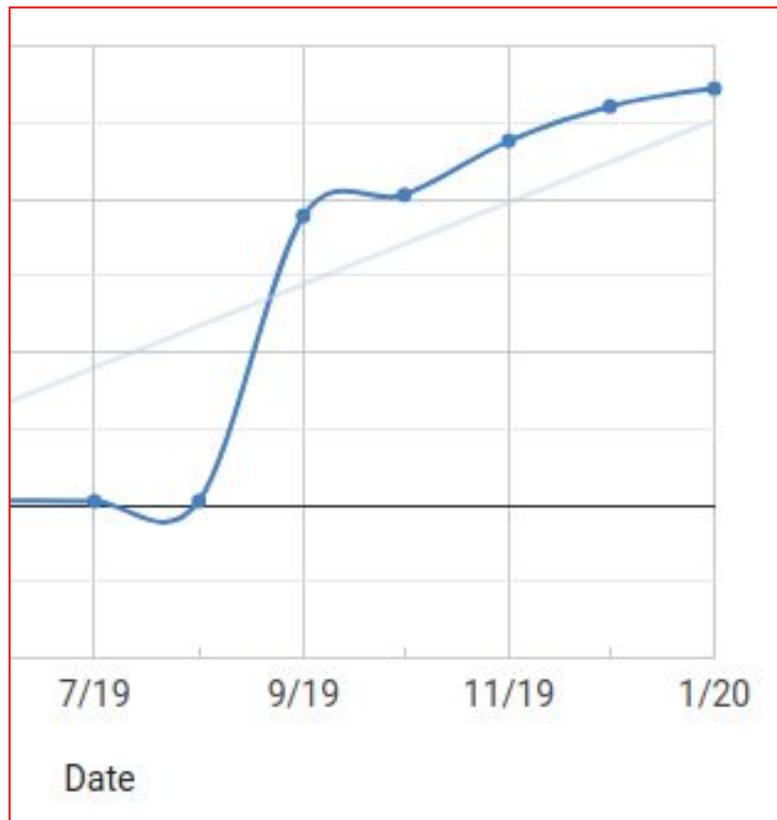
# Post-shipping work (1)

- Remove experimental flags.
- Continue conversations with WebPerf WG and eventually propose adopting the new API in the group.
- Address issues surfaced on GitHub repository.

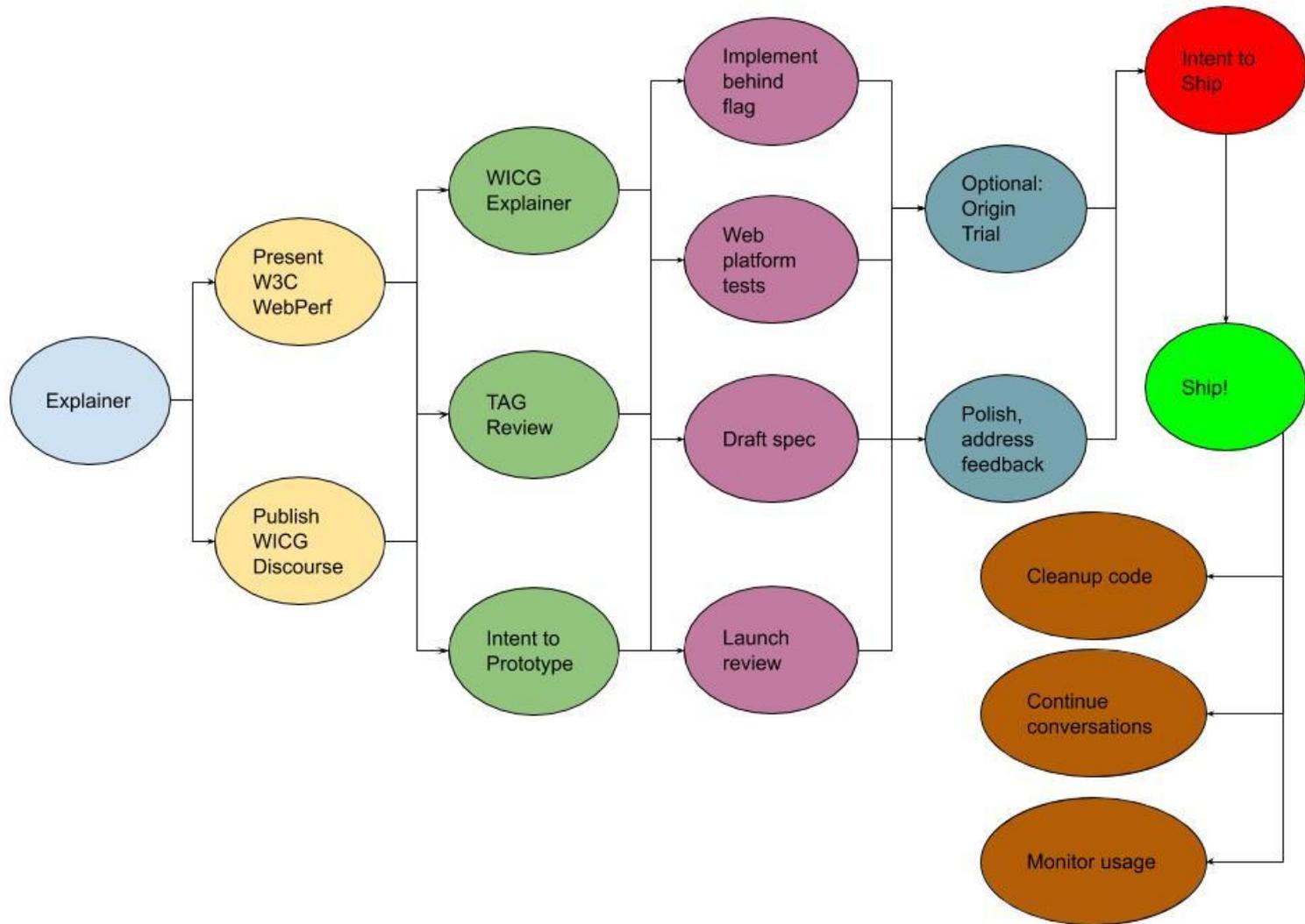


## Post-shipping work (2)

- Monitor usage and crashes
- We remove features that do not have multi-implementer support and have very little usage.



# Summary:



# Questions?

[npm@chromium.org](mailto:npm@chromium.org)

Twitter: @NicPenaM

GitHub: @npm1