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:

```html
<img src="hero.jpg" onload="performance.mark('herol')">  
<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/
Develop concrete proposal (1)

- Move explainer to WICG on GitHub. [https://github.com/WICG/element-timing](https://github.com/WICG/element-timing)
- Request design review from Technical Architecture Group (TAG).
Develop concrete proposal (2)

- Send Intent to Prototype (renamed from Implement).
Multiple Iterations

Updated version of proposal: https://docs.google.com/document/d/1sBM5lzDPws2mg1wRKiwM0TGFv9WgI6qEdF7vYhBYqUg/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This document is no longer being updated.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
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:

```html
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

```javascript
const observer = new PerformanceObserver(
    t.step_func_done(function(entryList) {
        assert_equals(entryList.getEntries().length, 1);
        const entry = entryList.getEntries()[0];
        const pathname = window.location.origin + '/element-timing/resources/circle.svg';
        checkElement(entry, pathname, 'my_svg', 'svg_id', beforeRender,
            document.getElementById('svg_id'));
        // Assume viewport has size at least 200, so the element is fully visible.
        checkRect(entry, [0, 200, 0, 200]);
        checkNaturalSize(entry, 200, 200);
    })
);
observer.observe({entryTypes: ['element']});
```
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 on Wed, Sep 12, 2018, 5:04 PM EDT

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, tdresser@chromium.org
Product owner: npm@chromium.org, 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

Contact emails
npm@chromium.org, tdresser@chromium.org

Spec
Explainer https://github.com/npm1/Element-Timing
More detailed doc: https://docs.google.com/document/d/1blFeMVdqxB0V3BAJh60ptOBFY7cJSXnf7VyW3wspbZ8/
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@
- 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 ✓
  #24 by npm1 was merged on Jul 24, 2019 • Approved

- Remove responseEnd section ✓
  #23 by npm1 was merged on Jul 24, 2019 • Approved

- Set identifier and id on entry creation ✓
  #22 by npm1 was merged on Jul 24, 2019 • Approved

- Define `elementTiming` as reflecting "elementtiming" ✓
  #19 by foolip was merged on Jul 24, 2019 • Approved

- Add loadTime to potentially-add-LCP ✓
  #17 by npm1 was merged on Jul 25, 2019 • Approved

- Update spec to use loadTime ✓
  #16 by npm1 was merged on Jul 23, 2019 • Approved

- Set rendering timestamp in renderTime ✓
  #15 by npm1 was merged on Jul 16, 2019 • Approved

- Add support for data URL images ✓
  #14 by npm1 was merged on Jul 12, 2019 • Approved

- Add hooks for LCP integration ✓
  #13 by yoaswales was merged on Jun 25, 2019 • Approved

- Remove implicit registration from spec ✓
  #10 by npm1 was merged on May 31, 2019

- Add text support ✓
  #7 by npm1 was merged on May 30, 2019

- Update image attributes ✓
  #6 by npm1 was merged on May 28, 2019

- Address feedback ✓
  #5 by npm1 was merged on May 24, 2019
<table>
<thead>
<tr>
<th>Subject</th>
<th>Status</th>
<th>Owner</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add buffered flag tests for UserTiming and ElementTiming</td>
<td>Merged</td>
<td>Nicolás Peña Moreno</td>
</tr>
<tr>
<td>(ElementTiming) Check opacity of image before exposing</td>
<td>Merged</td>
<td>Nicolás Peña Moreno</td>
</tr>
<tr>
<td>(ElementTiming) Add tests for generated and shadow text</td>
<td>Merged</td>
<td>Nicolás Peña Moreno</td>
</tr>
<tr>
<td>(ElementTiming) Add intersectionRect for text entries</td>
<td>Merged</td>
<td>Nicolás Peña Moreno</td>
</tr>
<tr>
<td>(ElementTiming) Do not walk objects that cannot be reported</td>
<td>Merged</td>
<td>Nicolás Peña Moreno</td>
</tr>
<tr>
<td>(ElementTiming) Text support part 2</td>
<td>Merged</td>
<td>Nicolás Peña Moreno</td>
</tr>
<tr>
<td>(ElementTiming) Change [name] and add [url] to PerformanceElementTiming</td>
<td>Abandoned, WIP</td>
<td>Nicolás Peña Moreno</td>
</tr>
<tr>
<td>(ElementTiming) Change [name] and add [url] to PerformanceElementTiming</td>
<td>Merged</td>
<td>Nicolás Peña Moreno</td>
</tr>
<tr>
<td>(ElementTiming) Add text support part 1</td>
<td>Merged</td>
<td>Nicolás Peña Moreno</td>
</tr>
<tr>
<td>(ElementTiming) Do not expose elements in Shadow Trees</td>
<td>Merged</td>
<td>Nicolás Peña Moreno</td>
</tr>
<tr>
<td>(ElementTiming) Add a test for invisible images</td>
<td>Merged</td>
<td>Nicolás Peña Moreno</td>
</tr>
<tr>
<td>(ElementTiming) Add element attribute</td>
<td>Merged</td>
<td>Nicolás Peña Moreno</td>
</tr>
<tr>
<td>(ElementTiming) Add test for image not added to DOM</td>
<td>Merged</td>
<td>Nicolás Peña Moreno</td>
</tr>
<tr>
<td>(ElementTiming) Fix DCHECK about natural width and height</td>
<td>Merged</td>
<td>Nicolás Peña Moreno</td>
</tr>
<tr>
<td>(ElementTiming) Add shadow image test</td>
<td>Merged</td>
<td>Nicolás Peña Moreno</td>
</tr>
<tr>
<td>(ElementTiming) Improve inline image behavior</td>
<td>Merged</td>
<td>Nicolás Peña Moreno</td>
</tr>
<tr>
<td>(ElementTiming) Add background image support</td>
<td>Merged</td>
<td>Nicolás Peña Moreno</td>
</tr>
<tr>
<td>(ElementTiming) Add element ID</td>
<td>Merged</td>
<td>Nicolás Peña Moreno</td>
</tr>
<tr>
<td>(ElementTiming) Add intrinsic size</td>
<td>Merged</td>
<td>Nicolás Peña Moreno</td>
</tr>
<tr>
<td>(ElementTiming) Mimic FCP++ rect computations</td>
<td>Merged</td>
<td>Nicolás Peña Moreno</td>
</tr>
<tr>
<td>(ElementTiming) Add carousel image test</td>
<td>Merged</td>
<td>Nicolás Peña Moreno</td>
</tr>
<tr>
<td>(ElementTiming) Expose cross origin elements that do not pass TAO</td>
<td>Merged</td>
<td>Nicolás Peña Moreno</td>
</tr>
<tr>
<td>(ElementTiming) Allow timing for video poster image</td>
<td>Merged</td>
<td>Nicolás Peña Moreno</td>
</tr>
<tr>
<td>(ElementTiming) Expose cross origin elements that do not pass TAO</td>
<td>Merged</td>
<td>Nicolás Peña Moreno</td>
</tr>
<tr>
<td>(ElementTiming) Add a test to prevent leak in ImageElementTiming</td>
<td>Merged</td>
<td>Nicolás Peña Moreno</td>
</tr>
</tbody>
</table>
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
Twitter: @NicPenaM
GitHub: @npm1