CSS/JS Coverage in Chrome DevTools

Since version 59, Chrome DevTools ship with a coverage analyzer that will tell you how much of the Javascript you load is actually used on your pages.

This is important because, if the code is yours, you can eliminate dead weight and reduce the size of your site's styles and scripts.

Running Coverage Analysis

Running coverage from Chrome DevTools is fairly straight forward. Open DevTools (Command + Option + I on Mac or Control + Shift + I on Windows) and look at the bottom of the DevTools window, there should be a `coverage` tab.

When you select the coverage tab you will see something like the figure below. Click on the reload icon to reload the page and analyze your site's coverage.

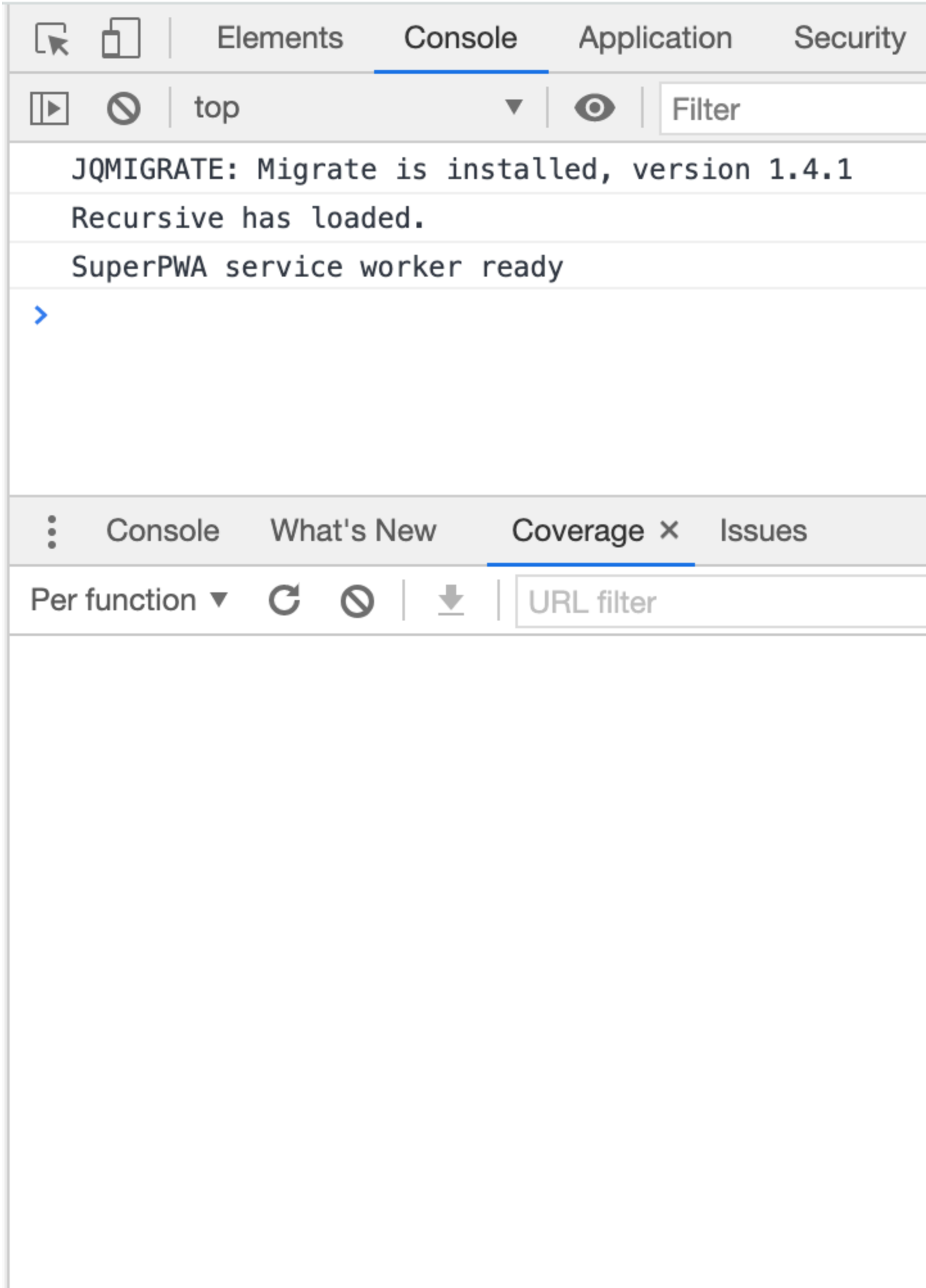
Elements
Console
Application
Security
top
Filter
JQMIGRATE: Migrate is installed, version 1.4.1
Recursive has loaded.
SuperPWA service worker ready
Console
What's New
Coverage
Issues
Per function
URL filter

Figure 1: Chrome DevTools showing coverage panel

The coverage panel will present a list of all the scripts and styles loaded on the page along with information about each resource loaded.

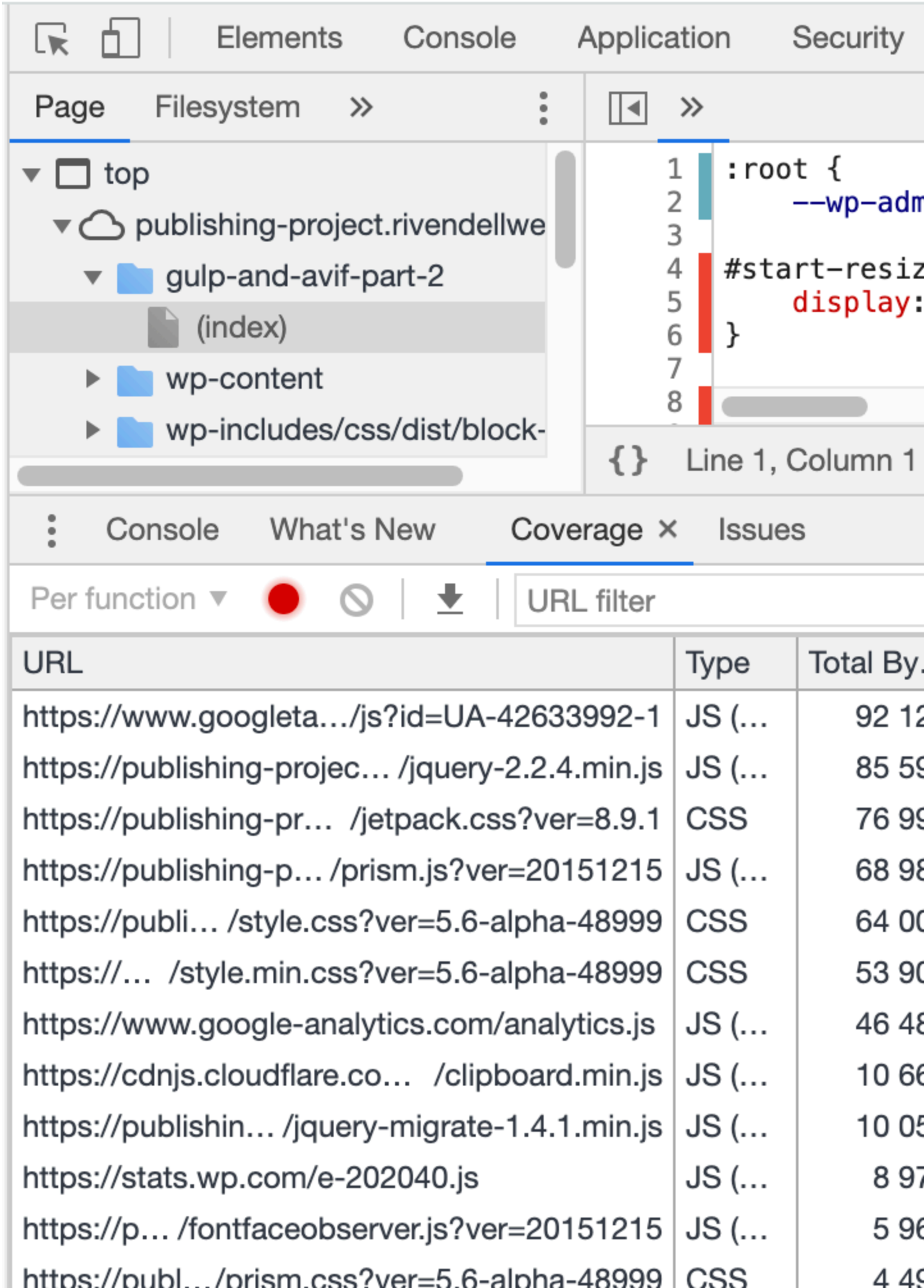
Elements
Console
Application
Security
Page
Filesystem
top
publishing-project.rivendellwe
gulp-and-avif-part-2
(index)
wp-content
wp-includes/css/dist/block-
:root {
--wp-adm
#start-resiz
display:
}
{}
Line 1, Column 1
Console
What's New
Coverage
Issues
Per function
URL filter
URL
Type
Total By
https://www.googleta…/js?id=UA-42633992-1
JS (…
92 12
https://publishing-projec… /jquery-2.2.4.min.js
JS (…
85 59
https://publishing-pr… /jetpack.css?ver=8.9.1
CSS
76 99
https://publishing-p… /prism.js?ver=20151215
JS (…
68 98
https://publi… /style.css?ver=5.6-alpha-48999
CSS
64 00
https://… /style.min.css?ver=5.6-alpha-48999
CSS
53 90
https://www.google-analytics.com/analytics.js
JS (…
46 48
https://cdnjs.cloudflare.co… /clipboard.min.js
JS (…
10 66
https://publishin… /jquery-migrate-1.4.1.min.js
JS (…
10 05
https://stats.wp.com/e-202040.js
JS (…
8 97
https://p… /fontfaceobserver.js?ver=20151215
JS (…
5 96
https://publ…/prism.css?ver=5.6-alpha-48999
CSS
4 49

Figure 2: Chrome DevTools showing the results of a coverage analysis

The infortmation provides is listed below:

- The **URL** of the resource that was analyzed
- The **Type** of resource (CSS, JavaScript, or both)
- The **Total Bytes** column shows the total size (in bytes) of the resource
- The **Unused Bytes** column is the number of unused bytes for the resource, both as a number of bytes and a percentage
- The last column shows a visualization of the Total Bytes and Unused Bytes columns
    - The red section of the bar is unused bytes
    - The green section is used bytes

Looking at the coverage table I can see that there are scripts and styles that are seldom used and can dig into what resources they are, whether I need them and whether I can reduce them further.

I've asked the devtools team on Twitter if it's possible to save the Coverage data as a CSV file to do further analysis offline.

Questions and Solutions

Unsurprisingly the largest resources are those I don't have direct access for modification. I need to do more research to see if I can tree shake these external scripts and how to do it.

The biggest question is how to shrink the size of these packages that I have no access to. Is it a matter of using a lighter version, like the slim package for jQuery, or is it a matter of using replacement libraries if they exist?

Doing it in CSS is easier. You can use UNCSS or similar tools to eliminate unused CSS. When working with CMSs like WordPress the question becomes: How do you do create and inline the Crtitical CSS in production for all pages on your site that we may create hourly or more frequently? Do we do it when the page is

visited?

This works best when you're building a site by hand or with a build system but how do you do it in a WordPress site or any other site that depends on a CMS?

As far as I know there is no free tool that will create critical CSS and inline it on pages and posts of a WordPress-powered site; most of them are parts of paid plugins. There are tools that have a PHP implementation of UNCSS, but there's still much research to do to see how it can be worked on a WordPress site. It scripts that need to remain in the head on their own. AMP plugins are the best/worst example of this behavior… they will fail if they are placed at the bottom of the page.