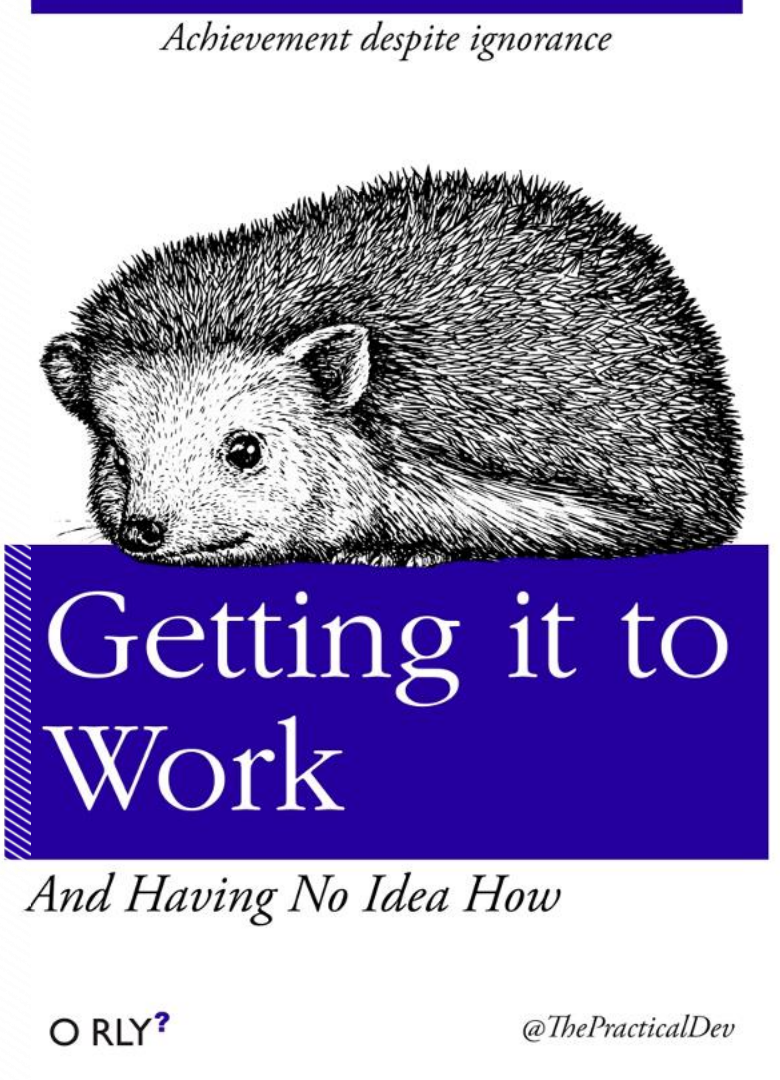
Debugging Tools

or how to leave console.log in prod code

Frontend Junior Program - 2021



Agenda

1. Intro
2. Debugging Tools
3. Chrome, Edge DevTools
4. Fiddler, Charles proxy
5. Nightly browser, extensions
6. Summary

INTRODUCTION

Preface

Working with JavaScript (even in projects), you **will have** this feeling:

Ahh, it is clear and easy!

ten minutes later …

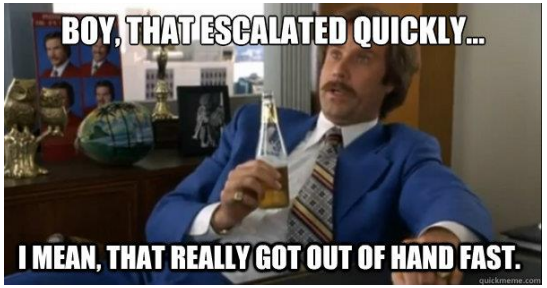
What the hell is going on?

A terrible mistake you can do at that point is that you try to make it work *somehow,* with several trial and error attempts.

At best, it won't work, and you just spent a lot of time on it.

At worst, it *will* work, and you certainly introduced a *bug* – and sometimes that will be very hard to find.

But don't worry, you don't need to do something wrong, bugs just born in their own rights.



"Debugging is twice as hard as writing the code in the first place. Therefore, if you write the code as cleverly as possible, you are, by definition, not smart enough to debug it."

The rubber duck method

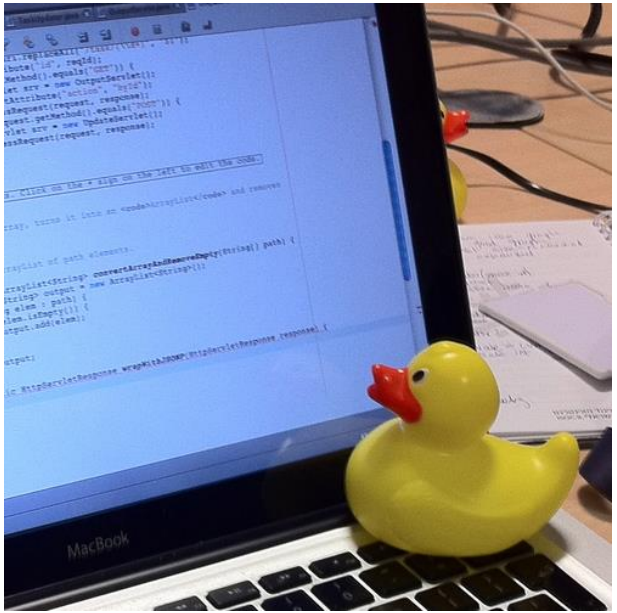
If you face with an issue, just slow down and try to fix it step by step - with understanding every detail of it.

Use the rubber duck method – it works!

Your colleagues also could play the role of the rubber duck. They often share tips and tricks, and it really helps to build relationship and think in teamwork.

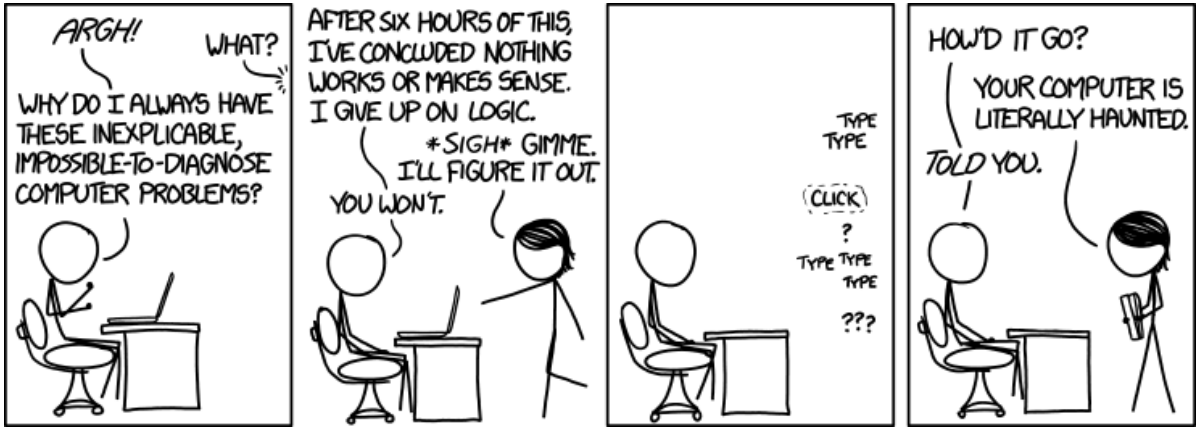
Also, it could really help to relieve your impostor syndrome.

While unit testing could help you to reason about the code, when the bundle reaches the browser, debugging tools will do the heavyweight in finding them.



this duck is a way stronger than you may think: it forces you to go through step by step (otherwise, you could skip seemingly trivial parts again and again)

What is debugging?



Debugging – is a process of search and elimination of errors in a code and providing information on the state of a system.

Debugging allows you to …

1. check the call stack
2. view the data
3. see the errors
4. override the data
5. monitor variables
6. add breakpoints

DEBUGGING TOOLS

Browser dev tools

Edge / Chrome

Built-in debugger based on Chromium

Firefox

Built-in debugger

Safari

Built in debugger
iOS debugger

Proxies

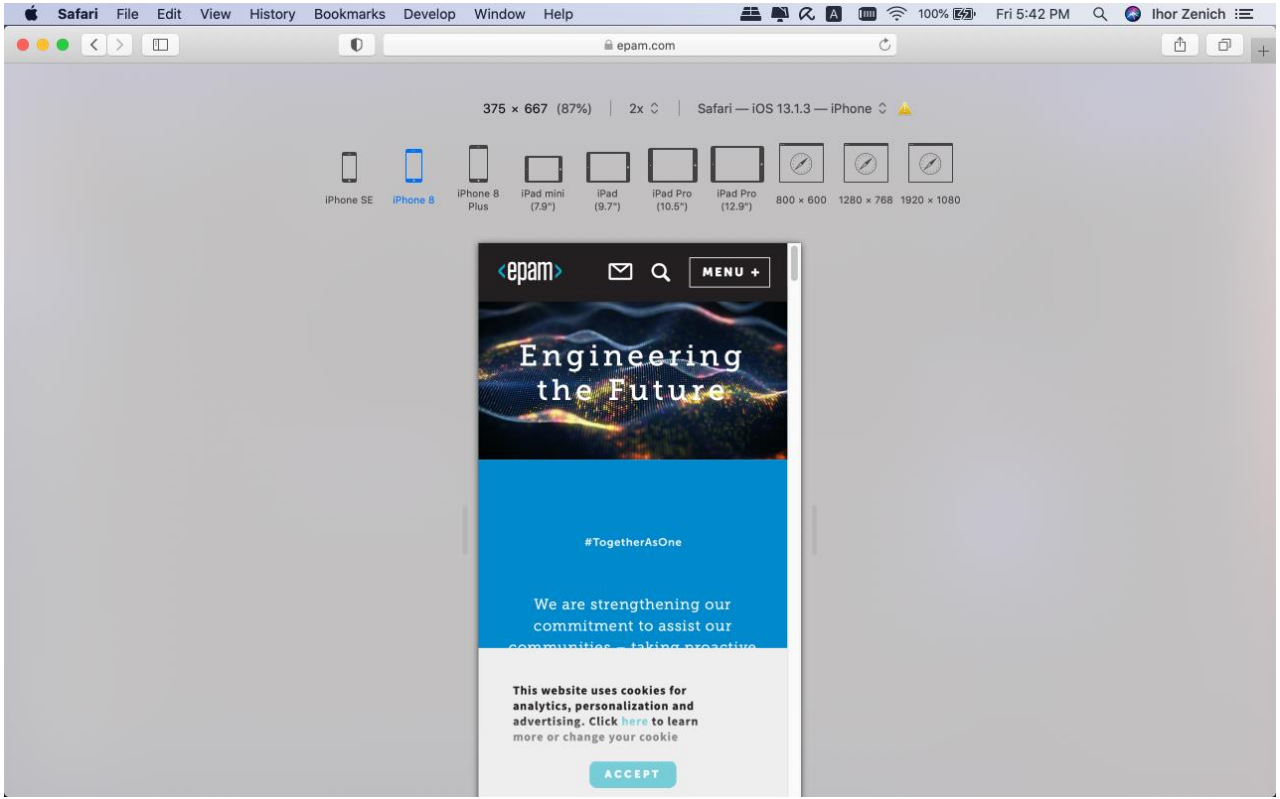
Charles is an HTTP proxy / HTTP monitor / Reverse Proxy that enables a developer to view all of the HTTP and SSL / HTTPS traffic between their machine and the Internet. This includes requests, responses and the HTTP headers (which contain the cookies and caching information).

In Web and Internet development you are unable to see what is being sent and received between your web browser / client and the server. Without this visibility it is difficult and time-consuming to determine exactly where the fault is. Charles makes it easy to see what is happening, so you can quickly diagnose and fix problems.

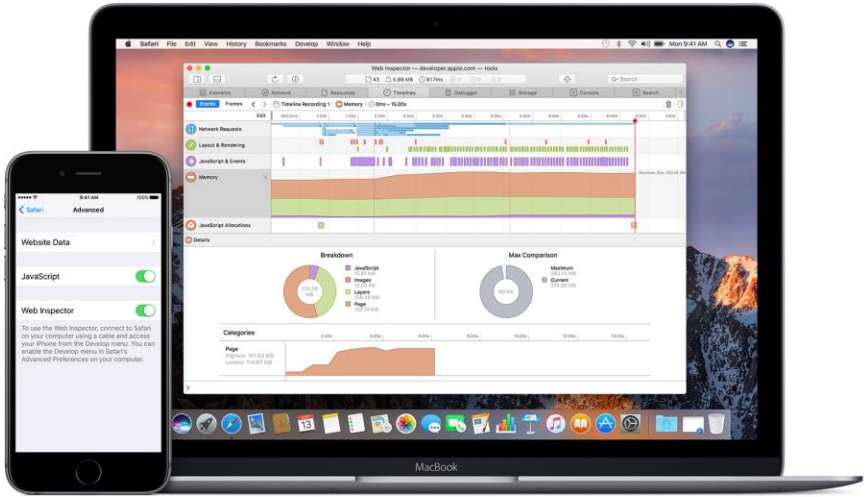
Charles proxy

Fiddler

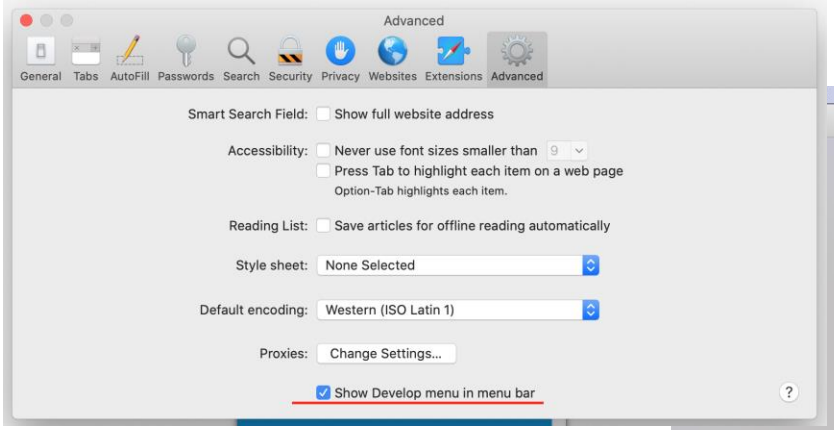
Safari Developer tools

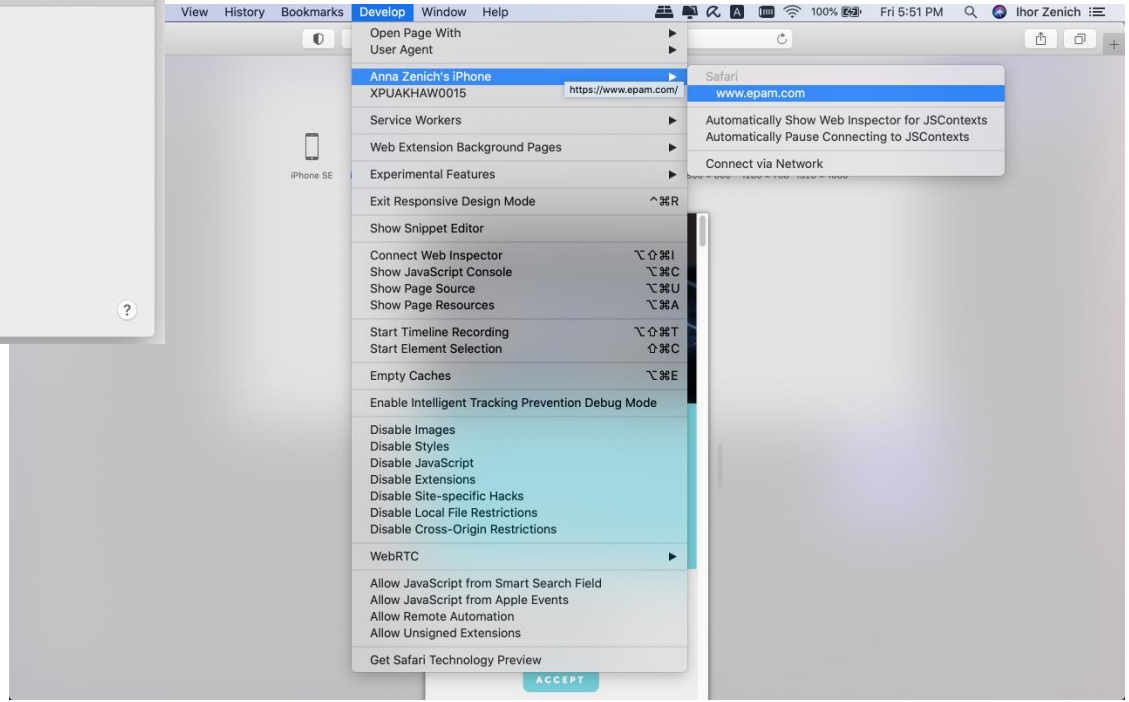


iOS devices remote debugging



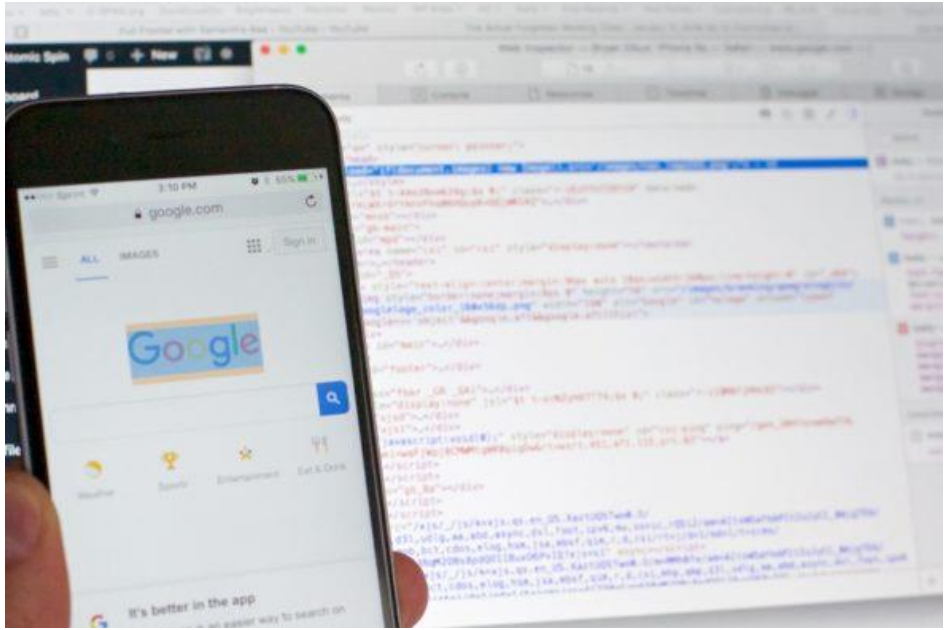
Safari – preferences (⌘,) and connecting a device





iOS devices remote debugging

After you connected an iOS device with Safari, remote debugging makes it possible to load a website on the device and inspect in the desktop Safari.

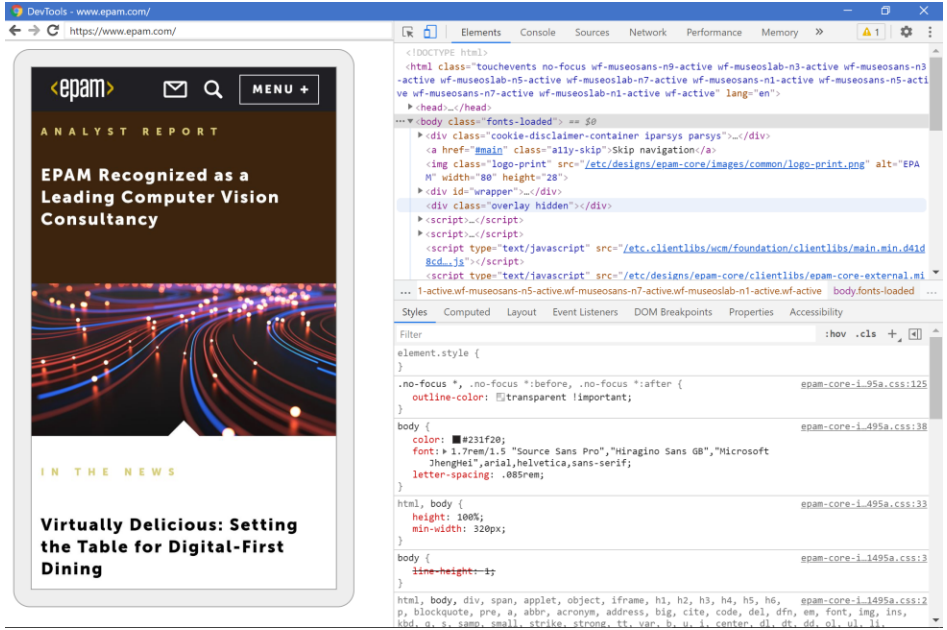


Android devices remote debugging

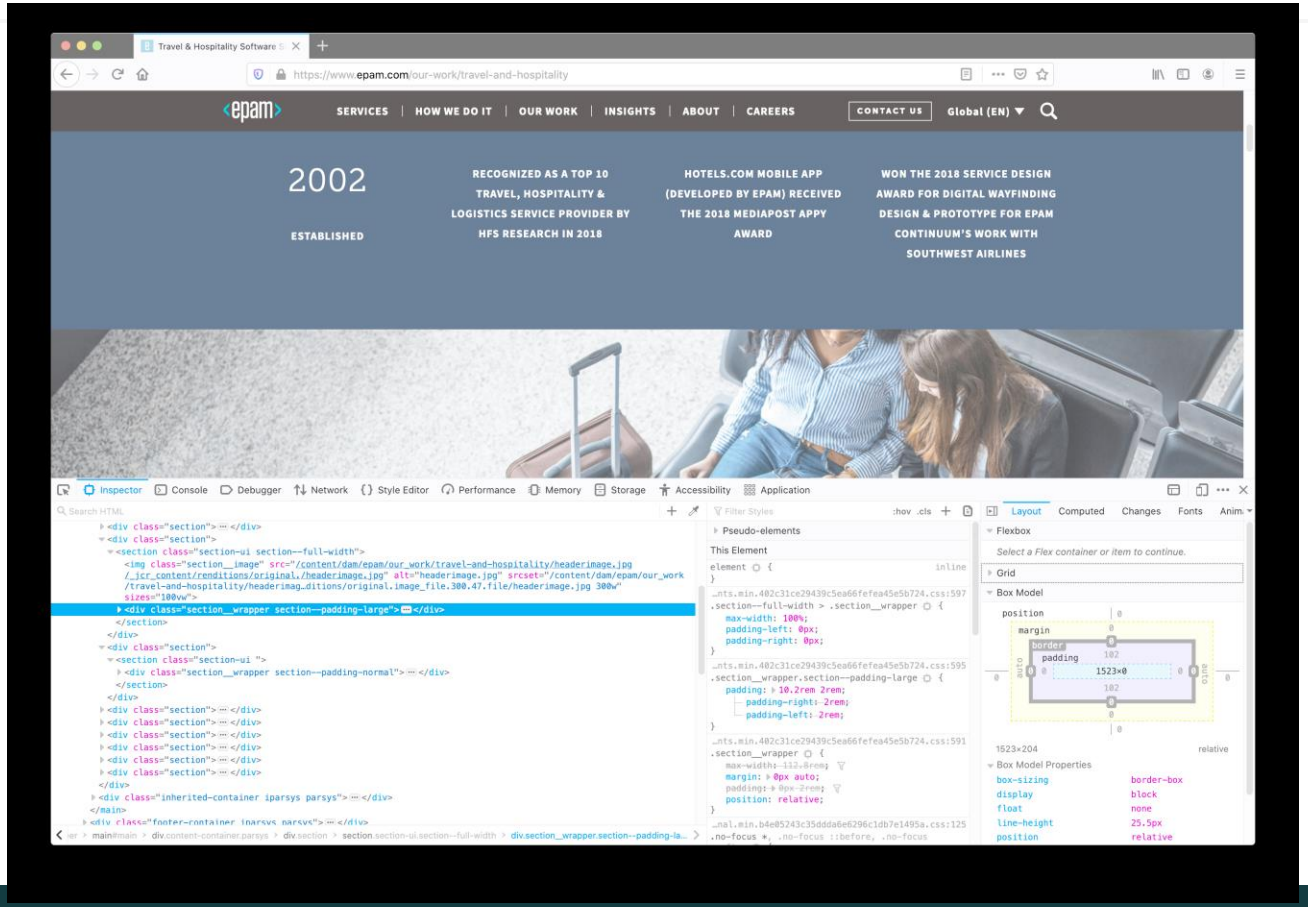
The same is possible with Android devices: both the desktop Chrome and Edge are capable to connect to an Android Chrome – and it works on Windows and on macOS as well.

Steps

1. connect the device with a USB cable
2. enable remote debugging on the phone
3. open chrome://inspect in the desktop browser (edge://inspect)
4. open Chrome on the Android device



Firefox Developer Tools



CHROME / EDGE DEVTOOLS

Chrome DevTools

Chrome DevTools

Main panels

Elements

Inspect and edit markup and styles

Network

Page-load performance

Sources

Debug with breakpoints

Application

Manage your local data

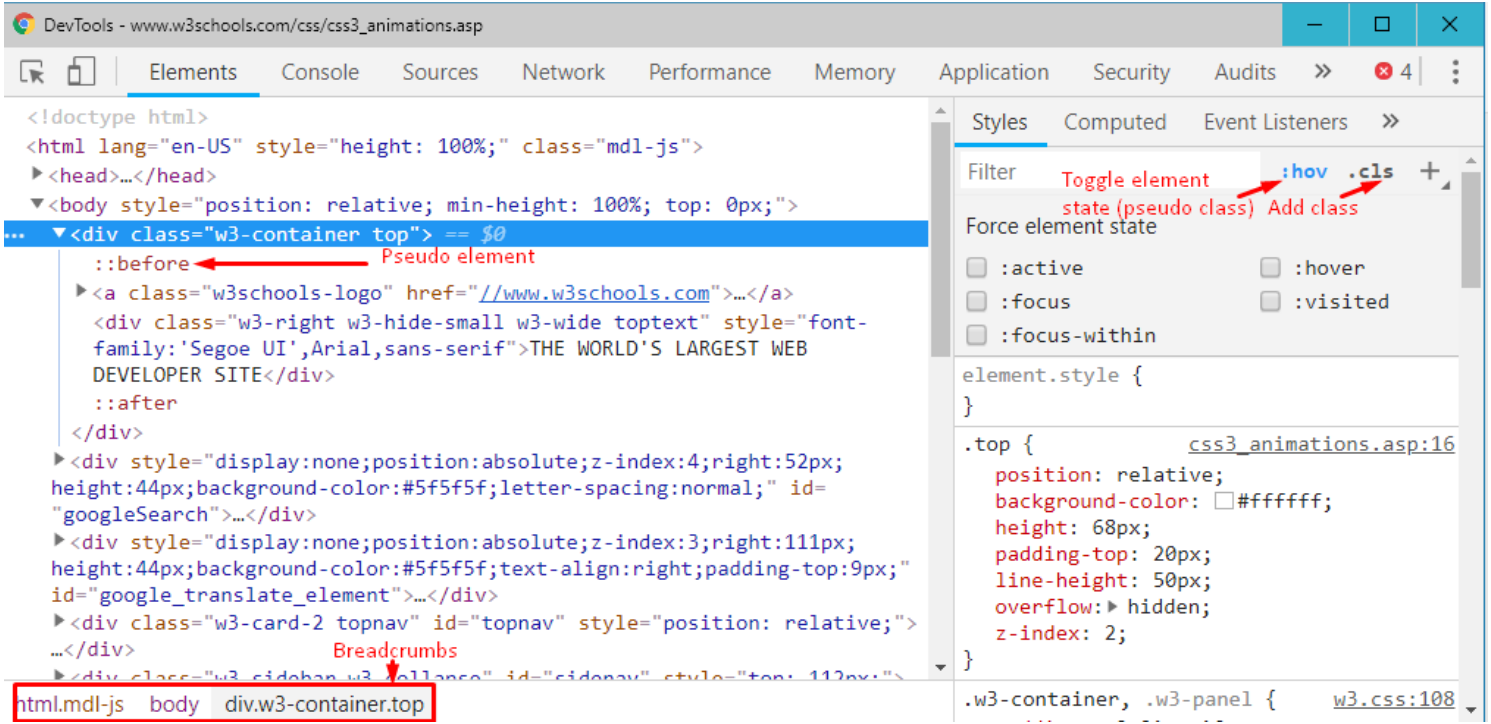
Lighthouse

Analyse a page as it loads

Console

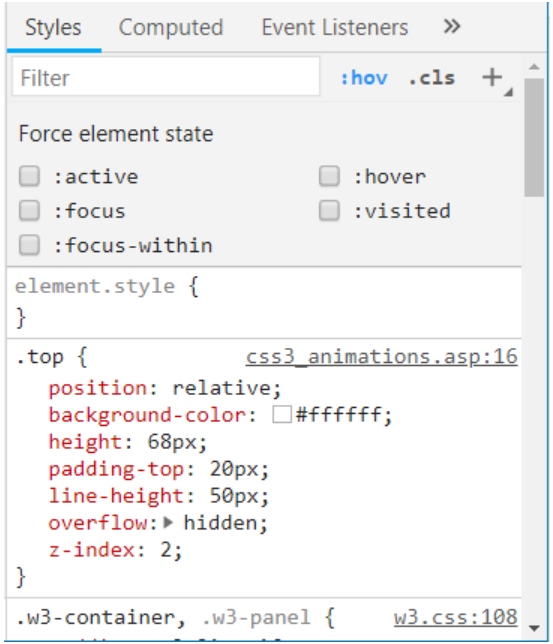
Interact from command Line

Elements

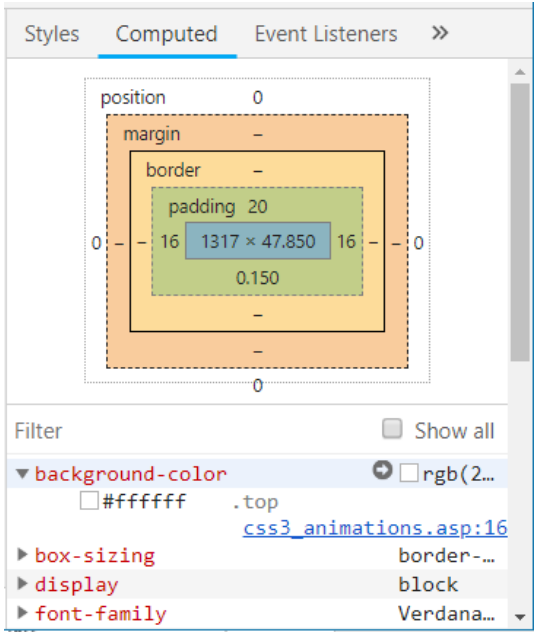

DevTools - www.w3schools.com/css/css3_animations.asp
Elements
Console
Sources
Network
Performance
Memory
Application
Security
Audits
Styles
Computed
Event Listeners
Filter
:hov .cls
Toggle element state (pseudo class)
Add class
Force element state
:active
:hover
:focus
:visited
:focus-within
Pseudo element
::before
::after
Breadcrumbs
html.mdl-js body div.w3-container.top
element.style {
.top {
css3_animations.asp:16
position: relative;
background-color: #ffffff;
height: 68px;
padding-top: 20px;
line-height: 50px;
overflow: hidden;
z-index: 2;
.w3-container, .w3-panel {
w3.css:108

Elements – Styles, Computed

Styles

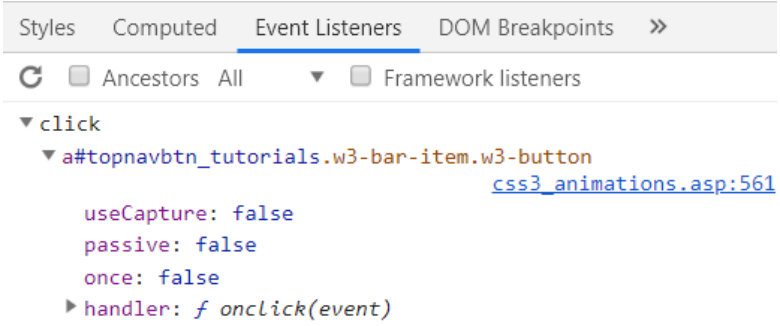


Computed



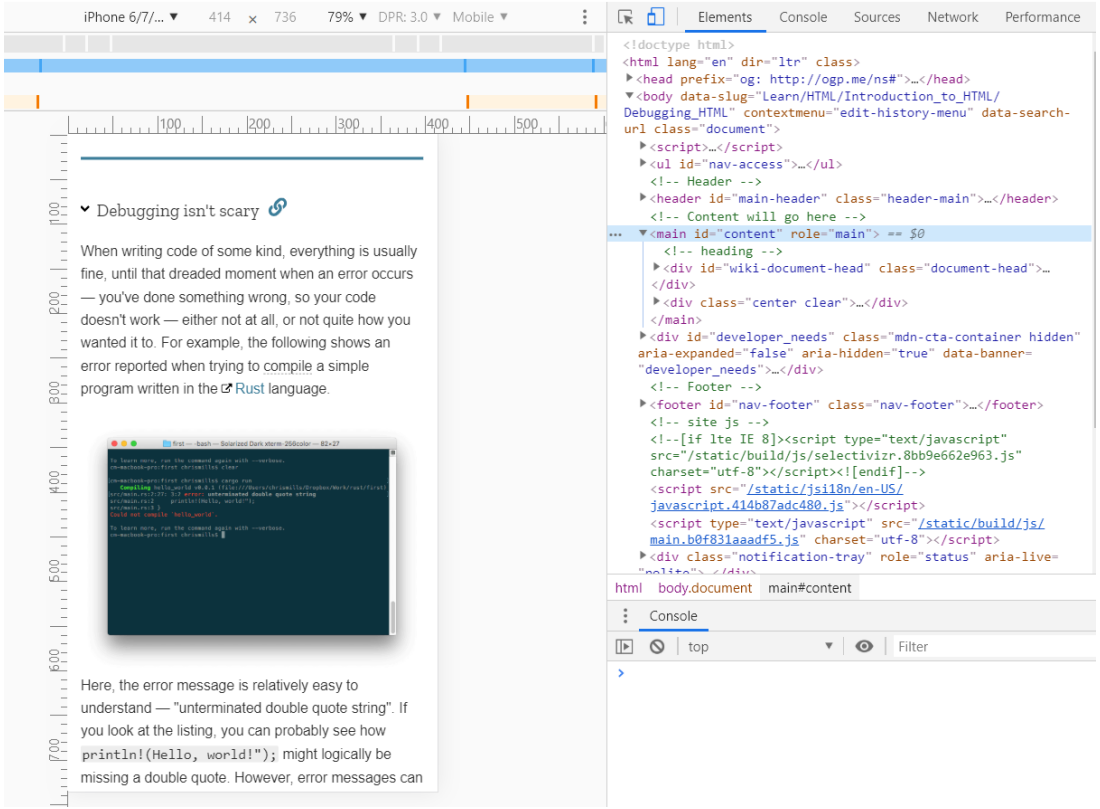
Elements - Event Listeners, DOM Breakpoints

Event Listeners

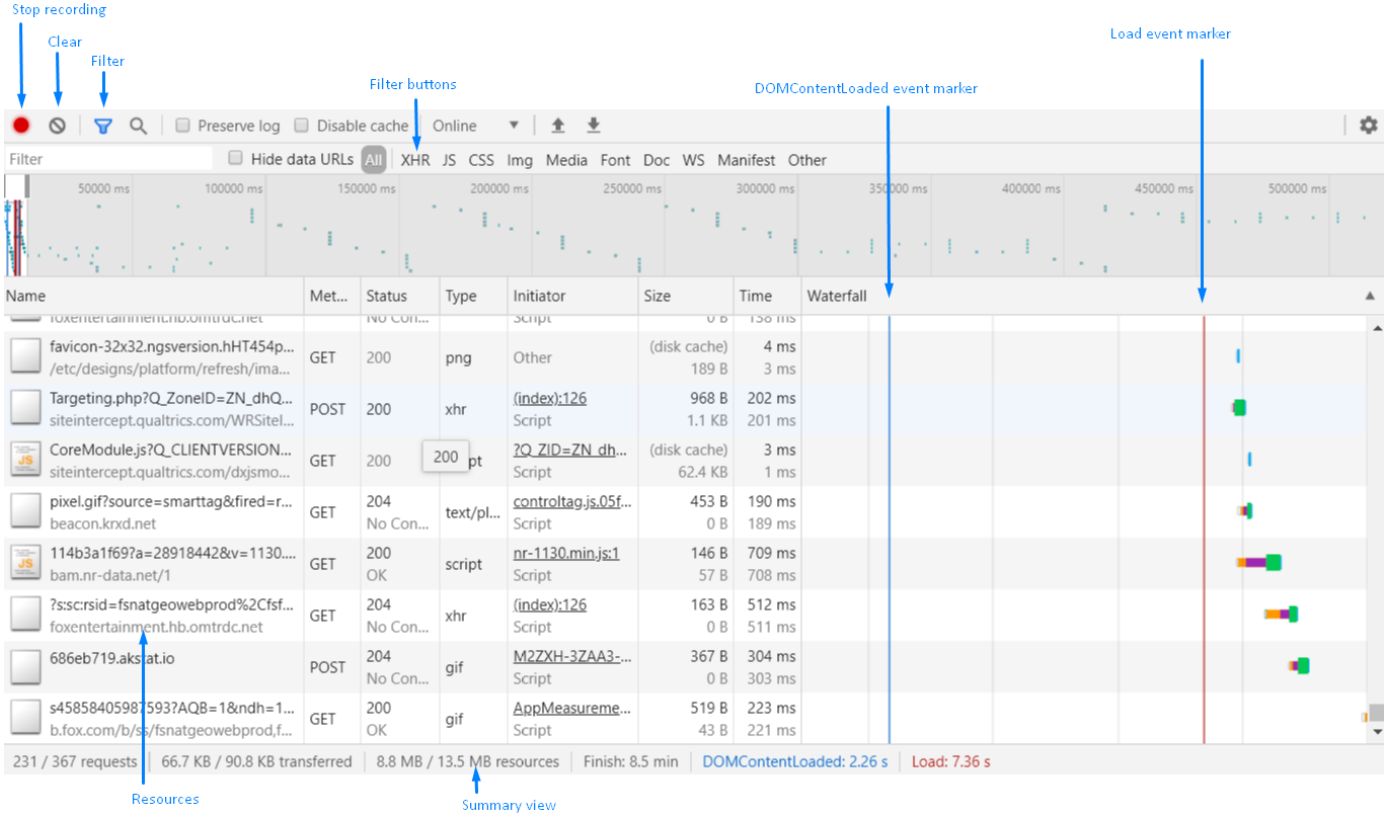


DOM Breakpoints

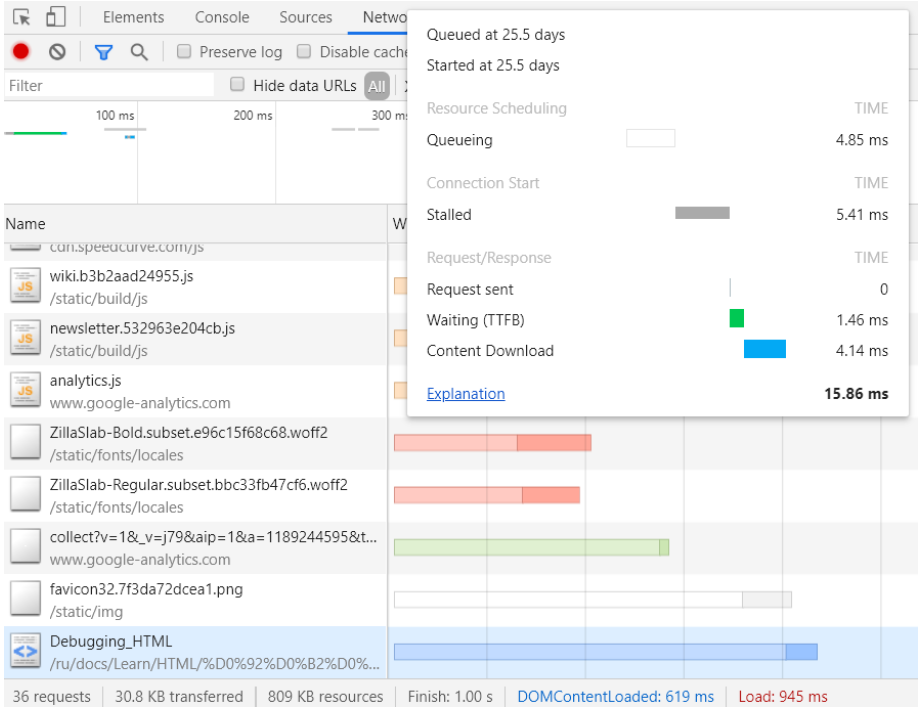
Viewport emulation



Network

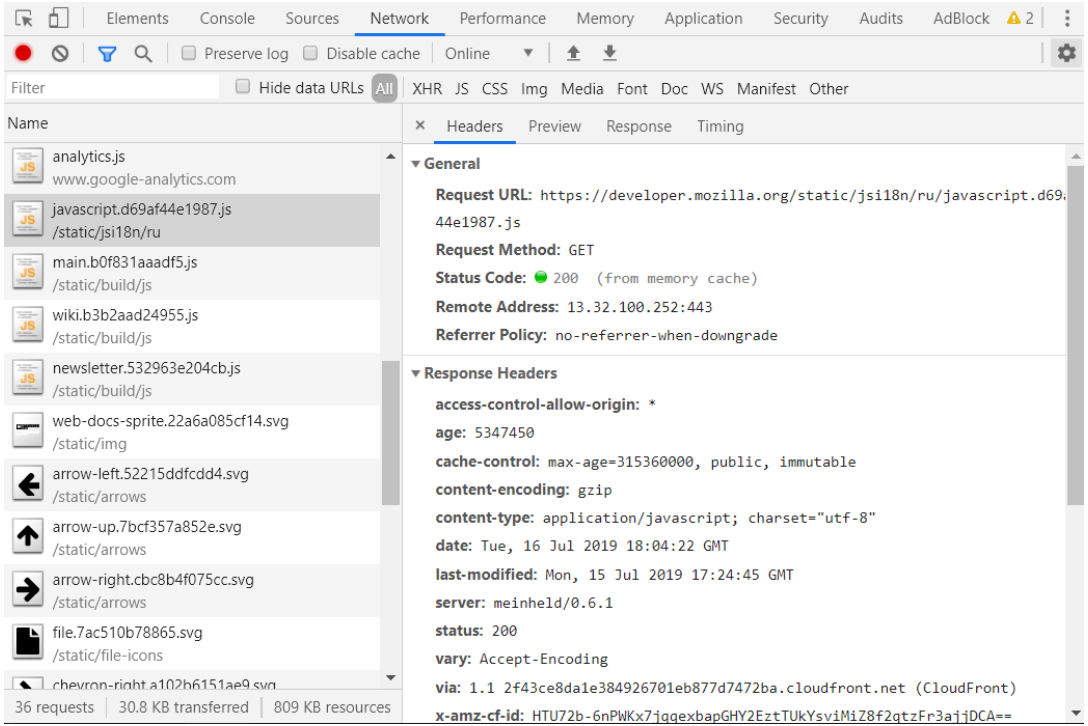

Stop recording
Clear
Filter
Filter buttons
DOMContentLoaded event marker
Load event marker
Preserve log
Disable cache
Online
Filter
Hide data URLs
All
XHR
JS
CSS
Img
Media
Font
Doc
WS
Manifest
Other
50000 ms
100000 ms
150000 ms
200000 ms
250000 ms
300000 ms
350000 ms
400000 ms
450000 ms
500000 ms
Name
Met...
Status
Type
Initiator
Size
Time
Waterfall
favicon-32x32.ngsversion.hHT454p...
/etc/designs/platform/refresh/ima...
GET
200
png
Other
(disk cache)
189 B
4 ms
3 ms
Targeting.php?Q_ZoneID=ZN_dhQ...
siteintercept.qualtrics.com/WRSitel...
POST
200
xhr
(index):126
Script
968 B
1.1 KB
202 ms
201 ms
CoreModule.js?Q_CLIENTVERSION...
siteintercept.qualtrics.com/dxjsmo...
GET
200
?Q_ZID=ZN_dh...
Script
(disk cache)
62.4 KB
3 ms
1 ms
200
pixel.gif?source=smarttag&fired=r...
beacon.krxd.net
GET
204
No Con...
text/pl...
controltag.js.05f...
Script
453 B
0 B
190 ms
189 ms
114b3a1f69?a=28918442&v=1130....
bam.nr-data.net/1
GET
200
OK
script
nr-1130.min.js:1
Script
146 B
57 B
709 ms
708 ms
?s:sc:rsid=fsnatgeowebprod%2Cfsf...
foxentertainment.hb.omtrdc.net
GET
204
No Con...
xhr
(index):126
Script
163 B
0 B
512 ms
511 ms
686eb719.aksstat.io
POST
204
No Con...
gif
M2ZXH-3ZAA3-...
Script
367 B
0 B
304 ms
303 ms
s45858405987593?AQB=1&ndh=1...
b.fox.com/b/ss/fsnatgeowebprod,f...
GET
200
OK
gif
AppMeasureme...
Script
519 B
43 B
223 ms
221 ms
231 / 367 requests
66.7 KB / 90.8 KB transferred
8.8 MB / 13.5 MB resources
Finish: 8.5 min
DOMContentLoaded: 2.26 s
Load: 7.36 s
Resources
Summary view

Network - data visualization



- Timing details visualization
- Load event markers
- Different sorting options:
  Start Time, Duration, Latency etc.

Network - Headers



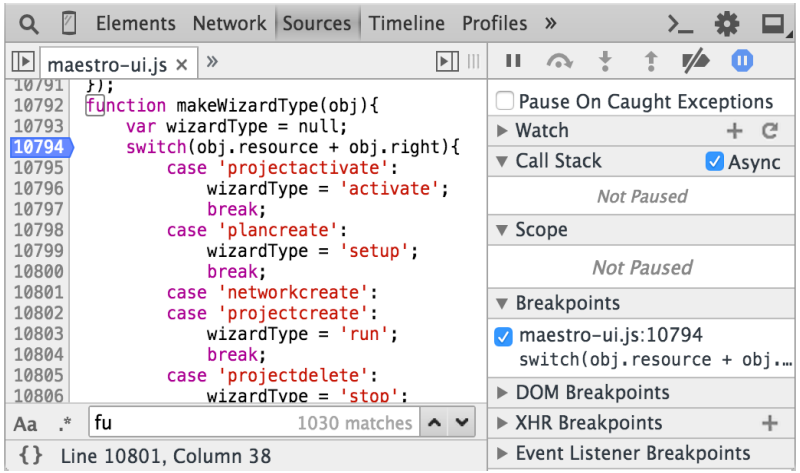
- Headers and payload
- Response body
- Cookies
- Timing

Sources - Breakpoints

Old School*

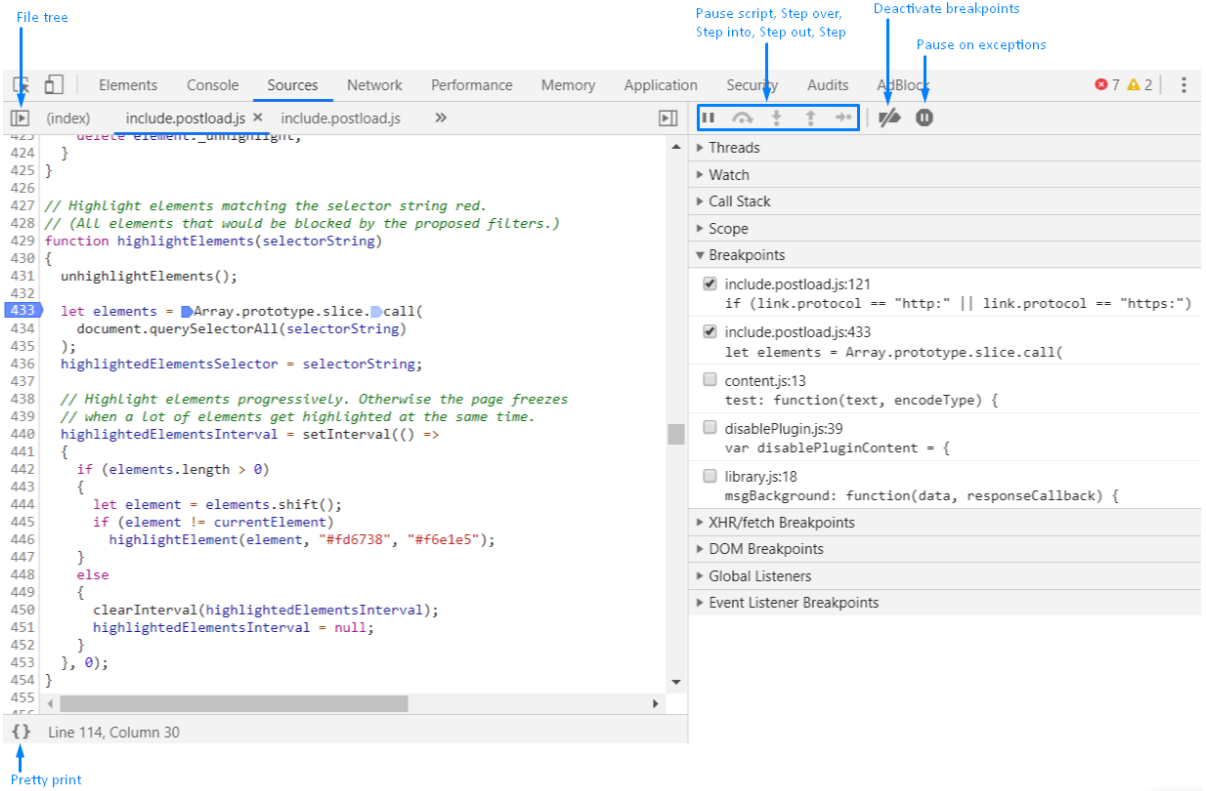
```
function makeWizardType(obj) {
    var wizardType = null;
    alert(obj.resource + obj.right);
    switch (obj.resource + obj.right) {
        case 'projectactivate':
            wizardType = 'activate';
            break;
        case 'plancreate':
            wizardType = 'setup';
            break;
        case 'networkcreate':
        case 'projectcreate':
            wizardType = 'run';
            break;
        case 'projectdelete':
            wizardType = 'stop';
            break;
    }
}
```

DevTools

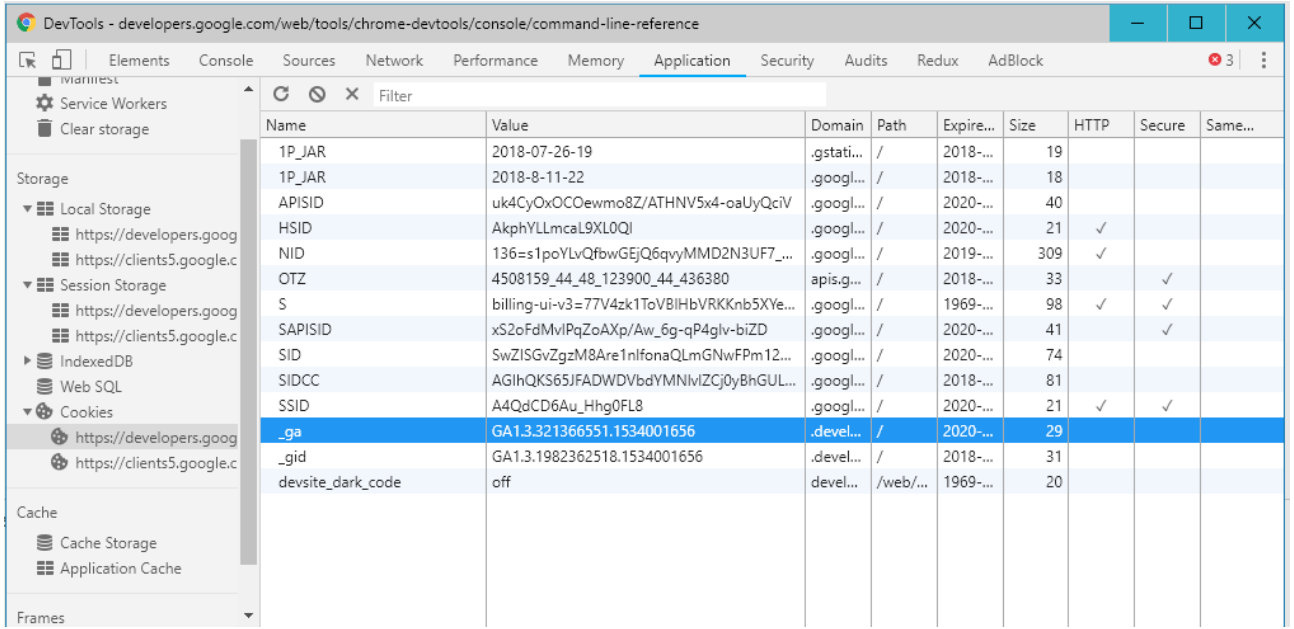


*"The most effective debugging tool is still careful thought, coupled with judiciously placed print statements."

Sources panel and debugging running code

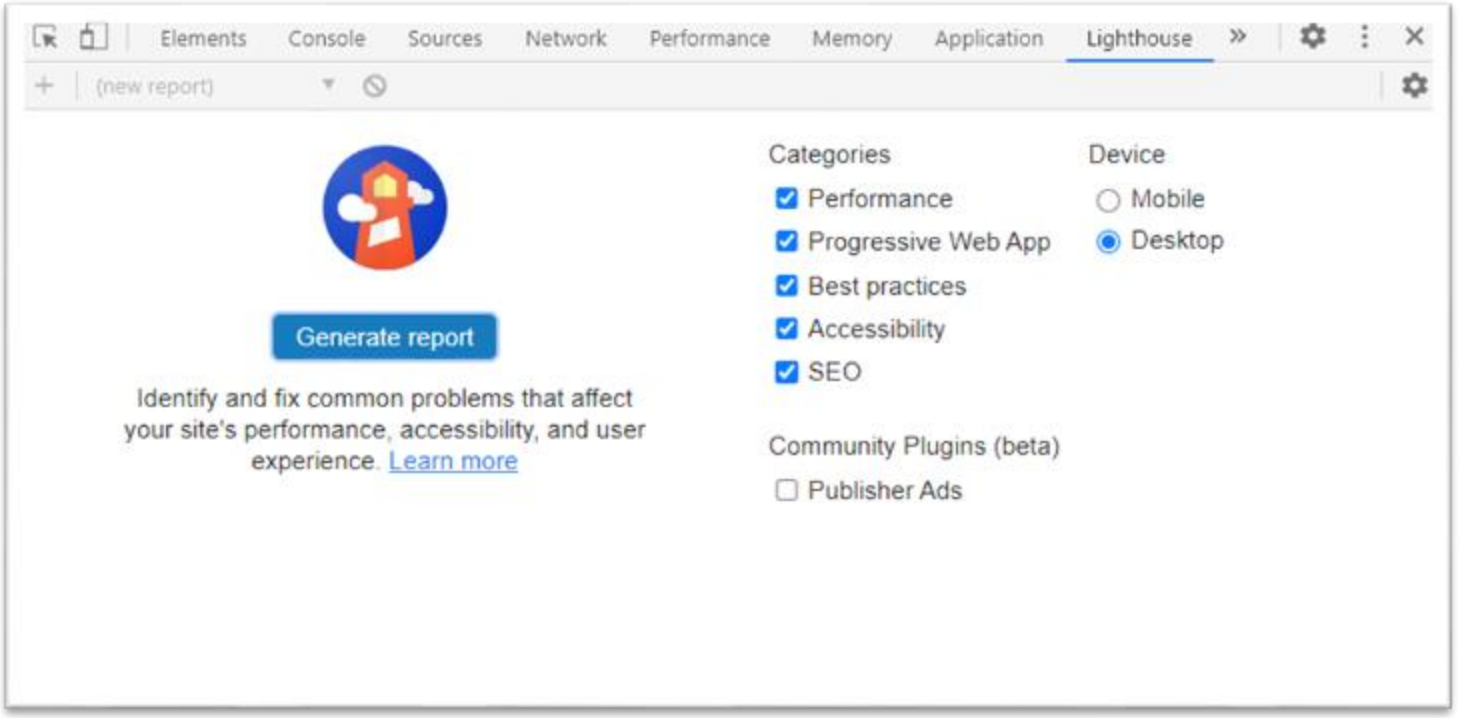


Application data

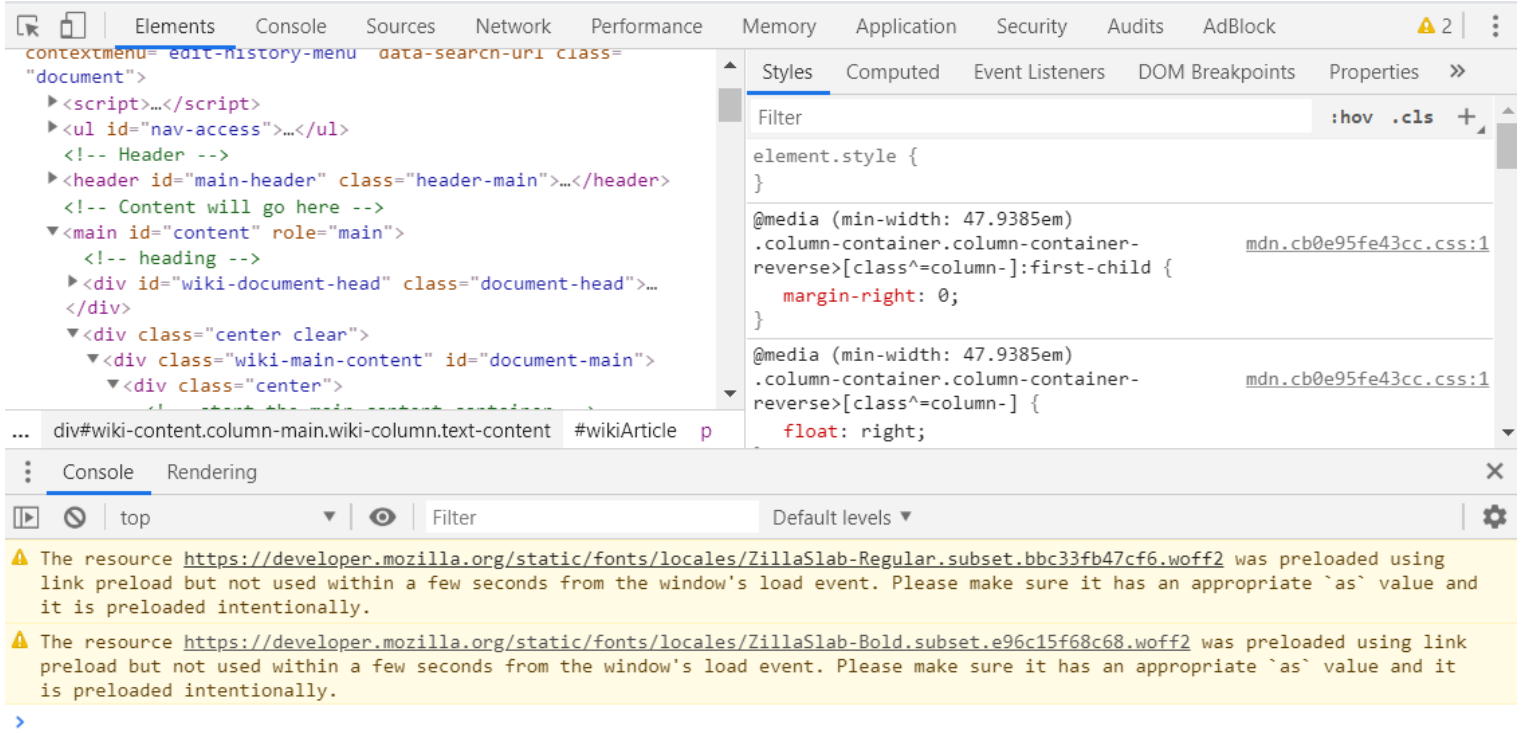


- HTML 5 Database
- Local Storage
- Cookies
- AppCache

Lighthouse

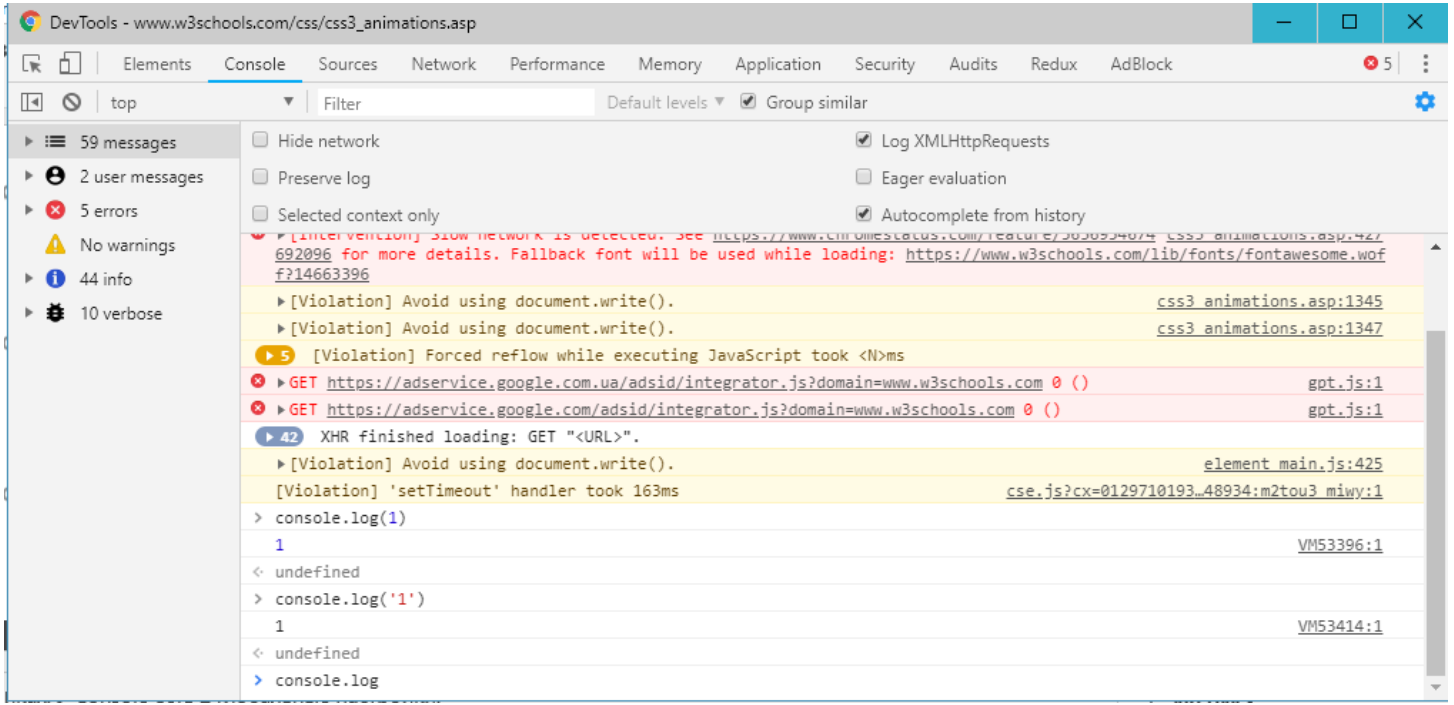

Elements
Console
Sources
Network
Performance
Memory
Application
Lighthouse
(new report)
Generate report
Identify and fix common problems that affect your site's performance, accessibility, and user experience. Learn more
Categories
Performance
Progressive Web App
Best practices
Accessibility
SEO
Device
Mobile
Desktop
Community Plugins (beta)
Publisher Ads

Console

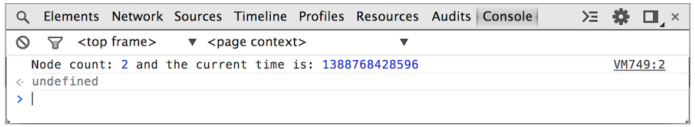

Elements
Console
Sources
Network
Performance
Memory
Application
Security
Audits
AdBlock
2
"document">
<script>…</script>
<ul id="nav-access">…
<!-- Header -->
<header id="main-header" class="header-main">…</header>
<!-- Content will go here -->
<main id="content" role="main">
<!-- heading -->
…

Styles
Computed
Event Listeners
DOM Breakpoints
Properties
Filter
:hov .cls
element.style {
}
@media (min-width: 47.9385em)
.column-container.column-container-reverse>[class^=column-]:first-child {
margin-right: 0;
}
mdn.cb0e95fe43cc.css:1
@media (min-width: 47.9385em)
.column-container.column-container-reverse>[class^=column-] {
float: right;
}
mdn.cb0e95fe43cc.css:1
div#wiki-content.column-main.wiki-column.text-content
#wikiArticle
p
Console
Rendering
top
Filter
Default levels
The resource https://developer.mozilla.org/static/fonts/locales/ZillaSlab-Regular.subset.bbc33fb47cf6.woff2 was preloaded using link preload but not used within a few seconds from the window's load event. Please make sure it has an appropriate `as` value and it is preloaded intentionally.
The resource https://developer.mozilla.org/static/fonts/locales/ZillaSlab-Bold.subset.e96c15f68c68.woff2 was preloaded using link preload but not used within a few seconds from the window's load event. Please make sure it has an appropriate `as` value and it is preloaded intentionally.

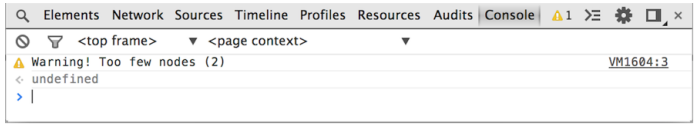
Console - Settings



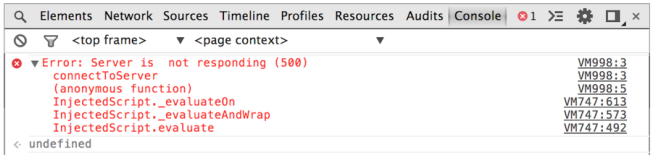
Console - API



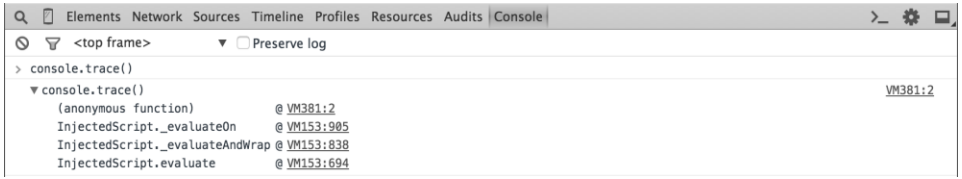
console.log()



console.warn()



console.assert()
console.error()

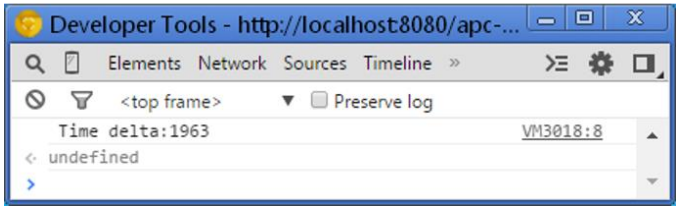


console.trace()

Console – Measuring running time

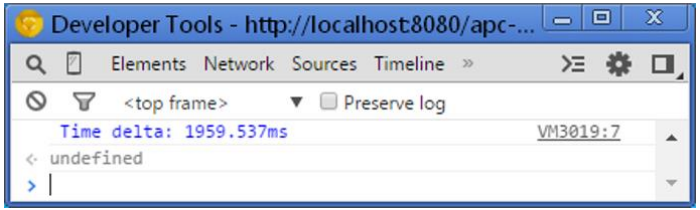
Old school

```
let t = new Date();
let array = new Array(1000000);
for (let i = array.length - 1; i >= 0; i--) {
    array[i] = {}
}
t = new Date() - t;
console.log('Time delta:' + t);
```

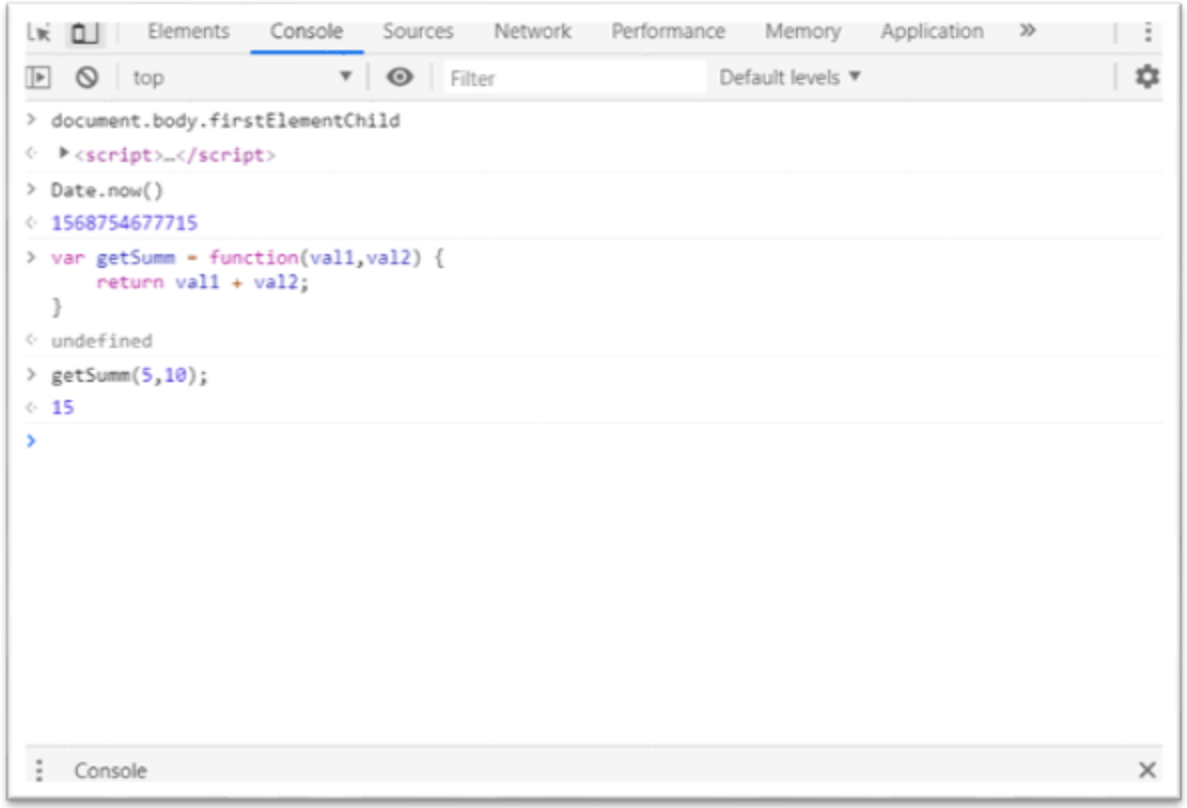


Console API

```
console.time('Time delta');
let array = new Array(1000000);
for (let i = array.length - 1; i >= 0; i--) {
    array[i] = {}
}
t = new Date() - t;
console.timeEnd('Time delta');
```



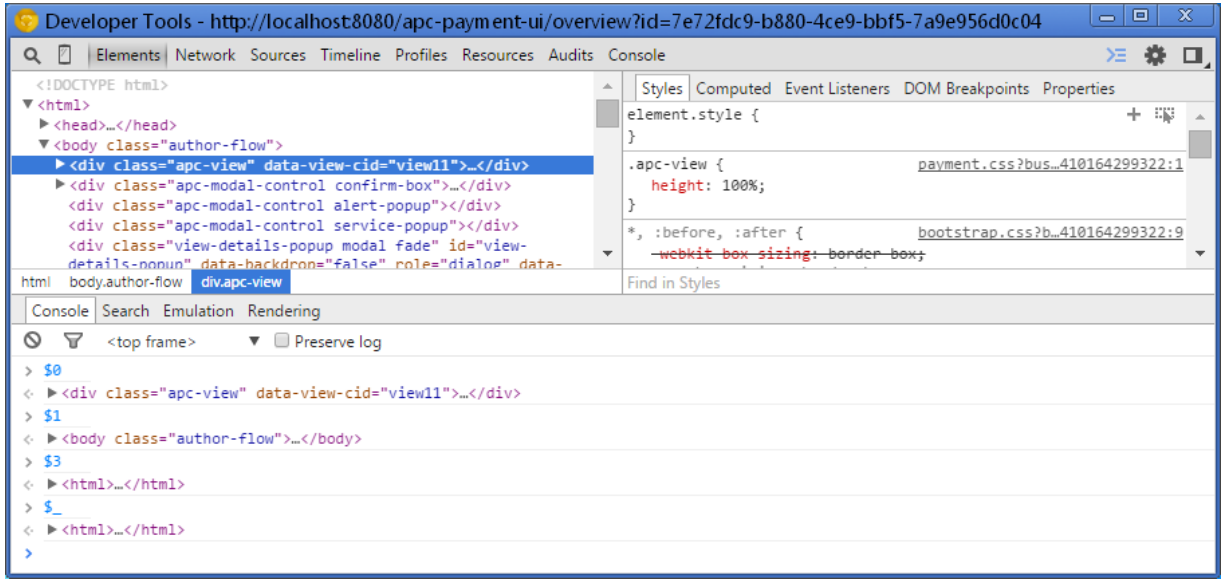
Console - Evaluating expressions



Console – Selecting elements

```
$0, $1, $2, $3, $4
$_
```

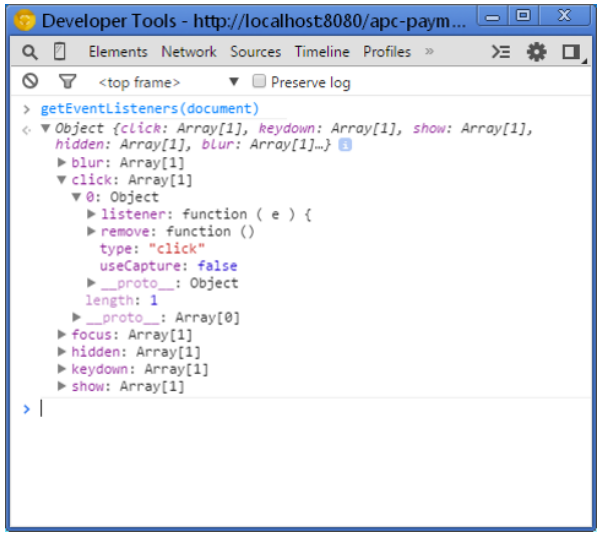
```
$('.article')
$$('.article')
```

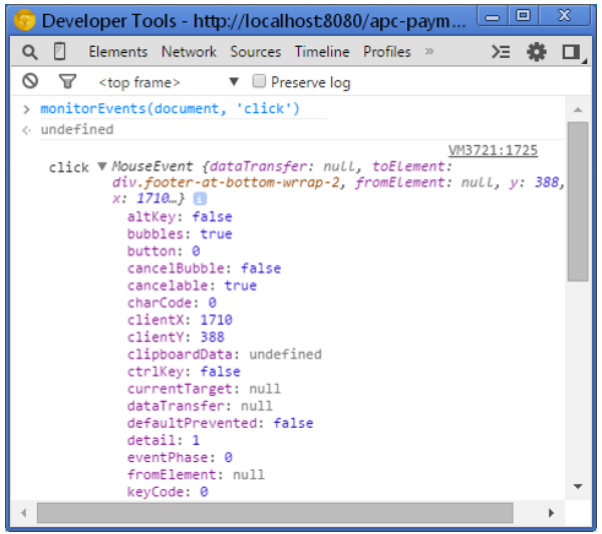


Console - Monitoring events

```
getEventListeners(object)
```

```
monitorEvents(window, "resize")
unmonitorEvents(object[, events])
```





Console - Utilities

Monitoring events

View event listeners registered on objects

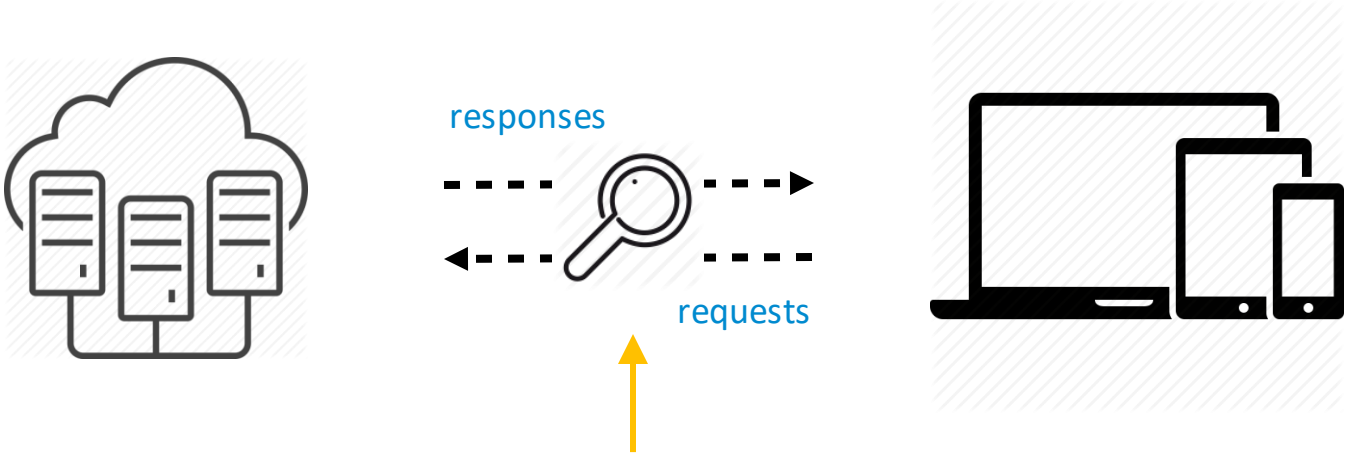
Aliases for a number of Console API methods

Evaluating JavaScript expressions

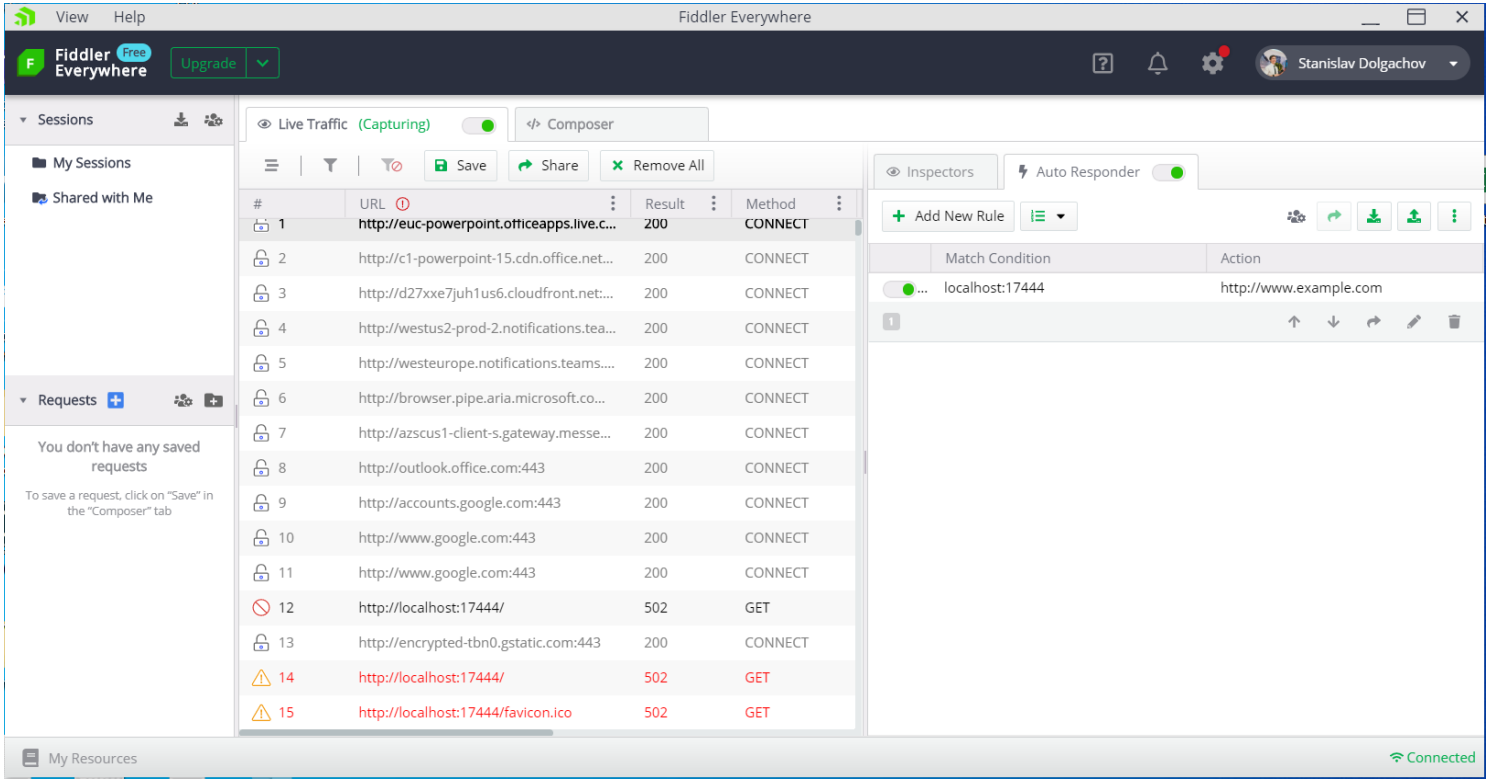
Convenience functions for selecting DOM elements

FIDDLER / CHARLES PROXY

Fiddler, Charles Proxy



Fiddler – AutoResponder



FIDDLER – LIST OF FEATURES

1. See all requests, headers, cookies and parameters transferred to / from the server.
2. Test the site on slow connection with the Internet
3. Rewrite transferred data / headers

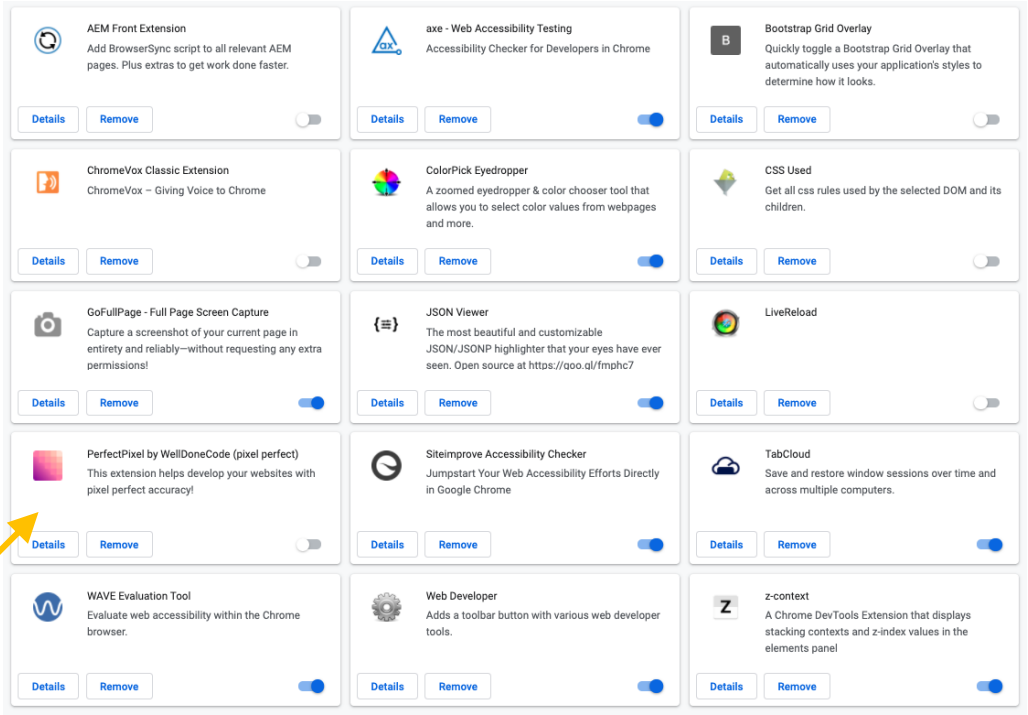
NIGHTLY BROWSERS, EXTENSIONS

Chrome extensions

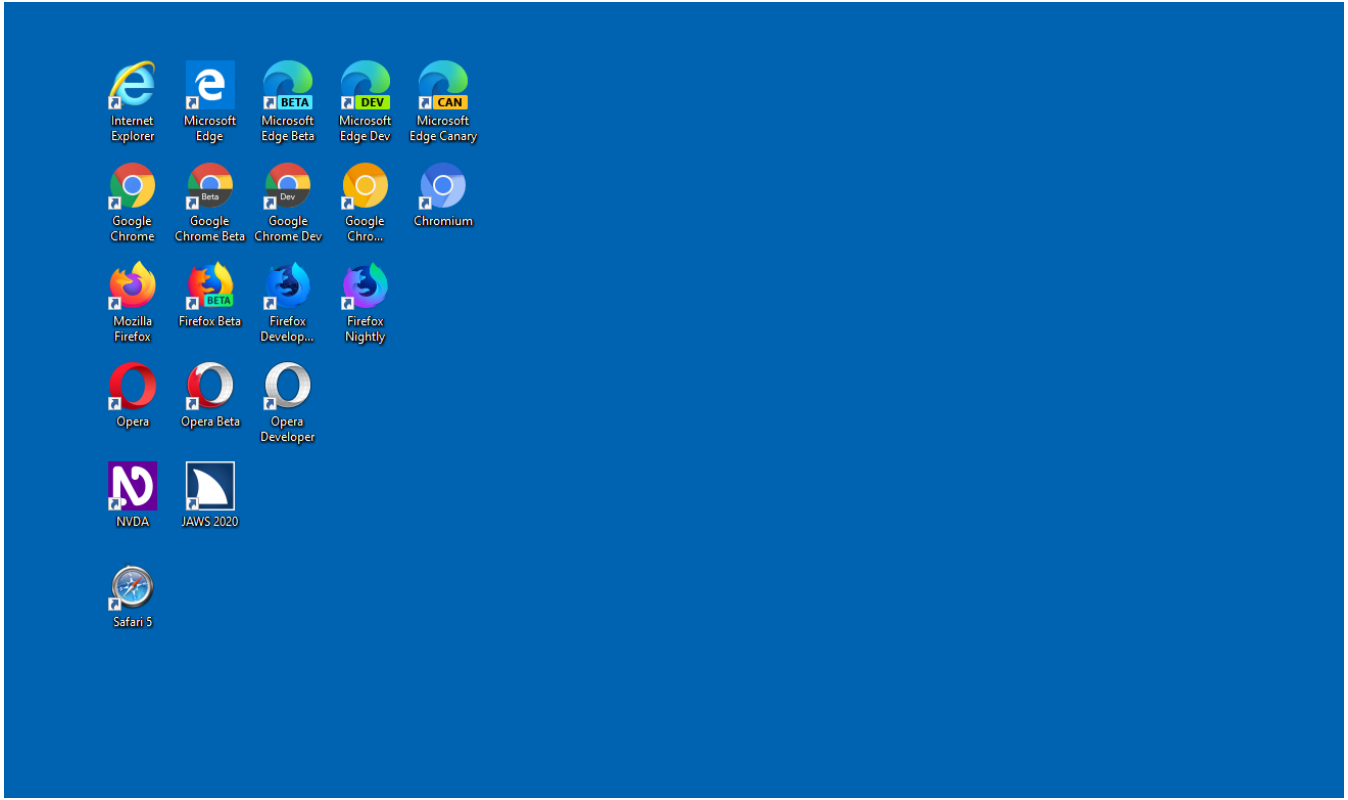
Always align extensions with your project policies!

Sometimes you simply cannot install extensions, sometimes it is possible, however, these can pose a security risk.

That one is really cool!



Nightly Browsers and Screen Readers




That's all Folks!

Debugging tools – wrap up

Easy To Use

Intuitively clear interface

See the errors

We can see errors during execution our code

Performance

Opportunity to check performance

State

Allows to see the state at the time of debugging

Accuracy

Pinpoints places and exact description of a problem

Monitoring

We can trace events

edu_hu@epam.com