epam
DEBUGGING
TOOLS
2021

AGENDA

INTRO

CHROME DEVTOOLS

PANELS AND SETTINGS

FIDDLER

SUMMARY

INTRODUCTION

WHAT IS DEBUGGING

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

DEBUGGING ALLOWS YOU TO SEE

1. Current call stack
2. Data of elements
3. See the errors
4. Override the data
5. Monitor variables
6. Adding breakpoints

TYPE LIST OF TOOLS FOR DEBUGGING AND PROFILING

DEV TOOLS: MOST USEFUL AND POPULAR DEVELOPER TOOLS

Chrome

Built-in debugger of Chrome

Firefox

Built-in debugger of Firefox

Internet Explorer

Built in debugger of Internet Explorer

Fiddler

Proxy which logs all HTTP(S)

Firebug

Expansion debugging for Firefox

Visual Events

Extension for visualization of events

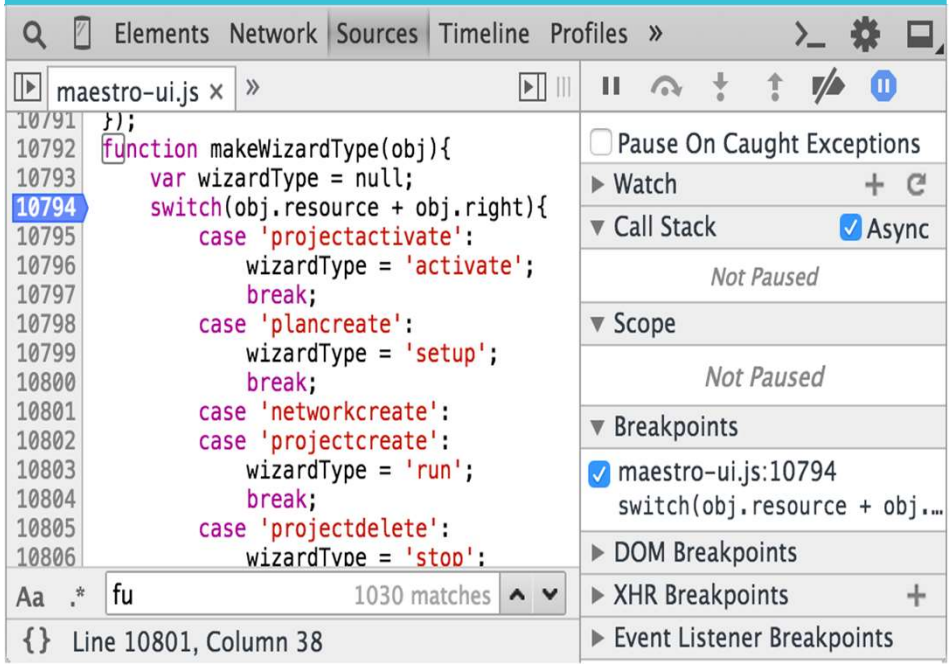
CHROME VS IE VS FIREFOX

Instruments	Chrome	IE	Firefox
Elements	✓	✓	✓
Network	✓	✓	✓
Sources	✓	✓	✓
Console	✓ 2X	✓	✓
Emulation	✓ 2X	✓	✓

PROBLEMS VISUALLY

Debugging of the healthy person

JAVASCRIPT



Debugging of the smoker

JAVASCRIPT

```
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;
```

chrome
DEVELOPER TOOLS

CHROME PANELS

Elements
Inspect and Edit Pages and Styles

Network
Page Load Performance

Sources
Set Up Your Workspace Debug with Breakpoints

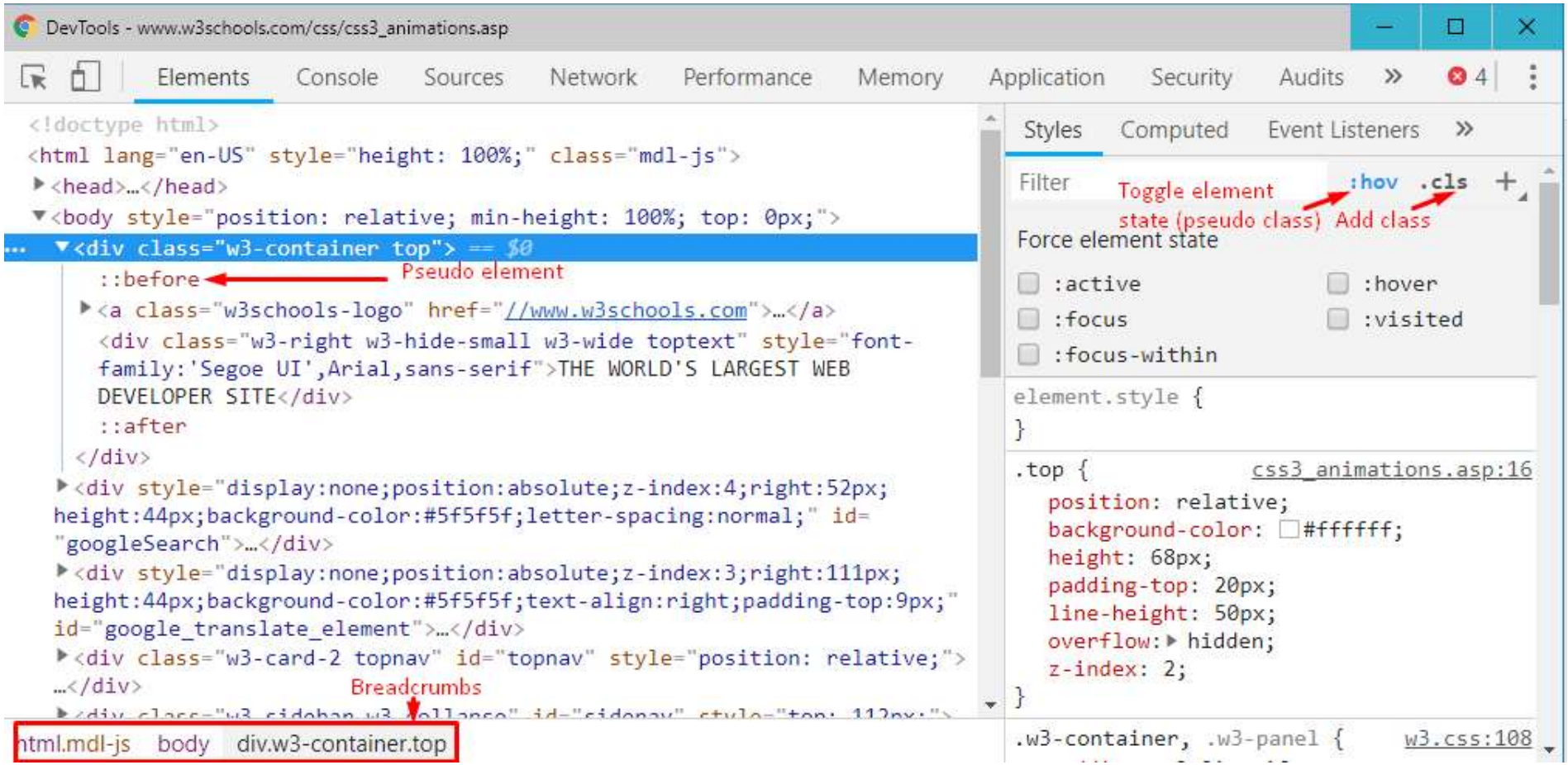
Application
Manage Your Local Data Resources

Audits
Analyze a page as it loads

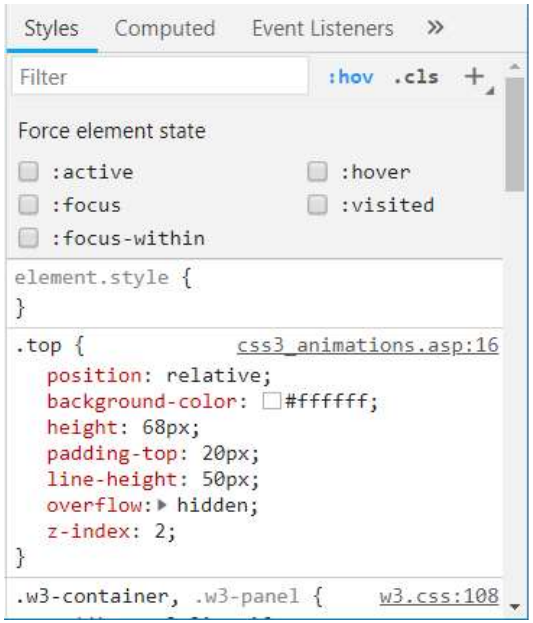
Console
Using the Console Interact from Command Line

ELEMENTS

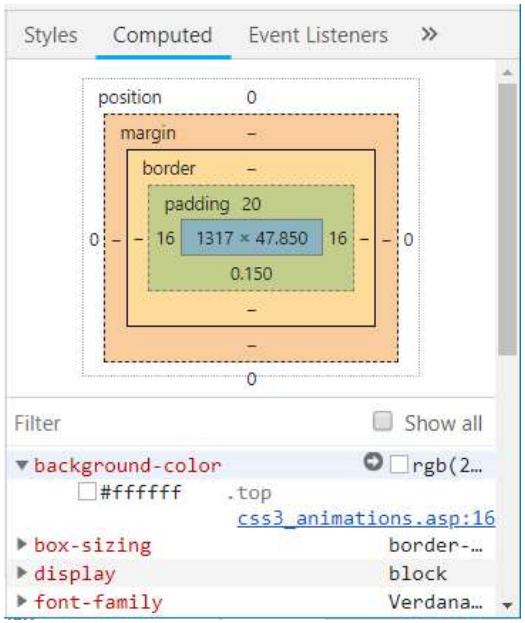
ELEMENTS PANEL



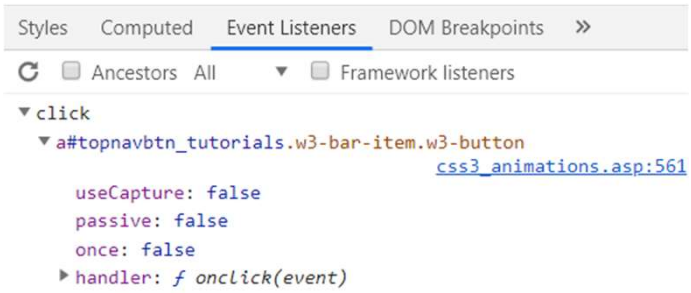
Styles



Computed



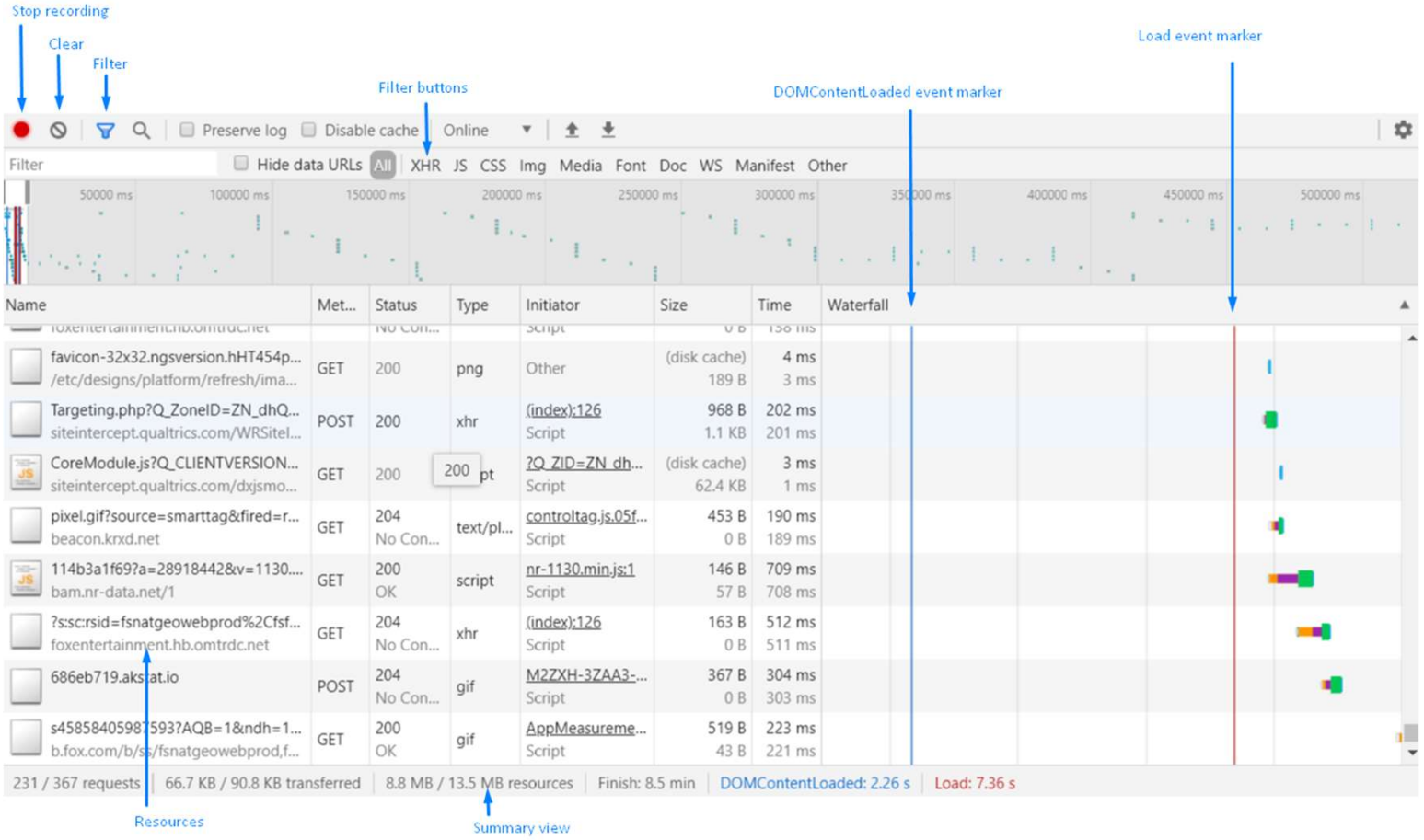
Event Listeners



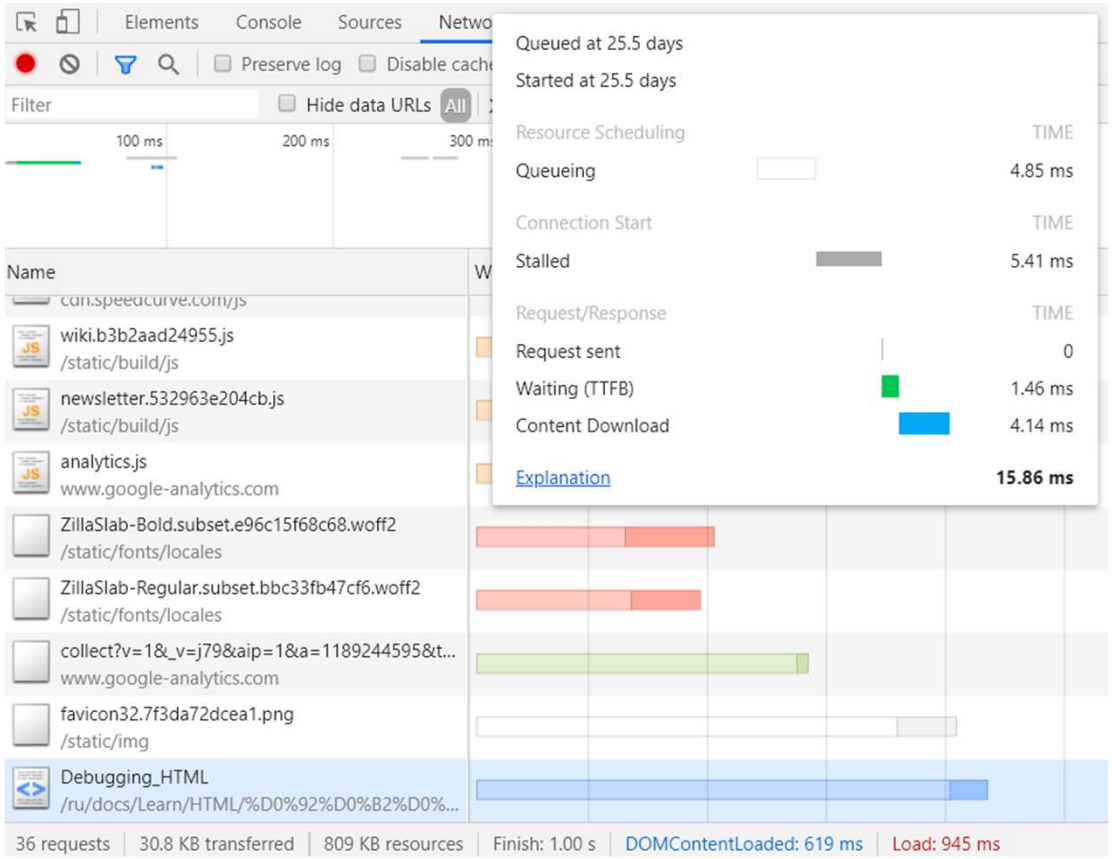
DOM Breakpoints

NETWORK

NETWORK PANEL



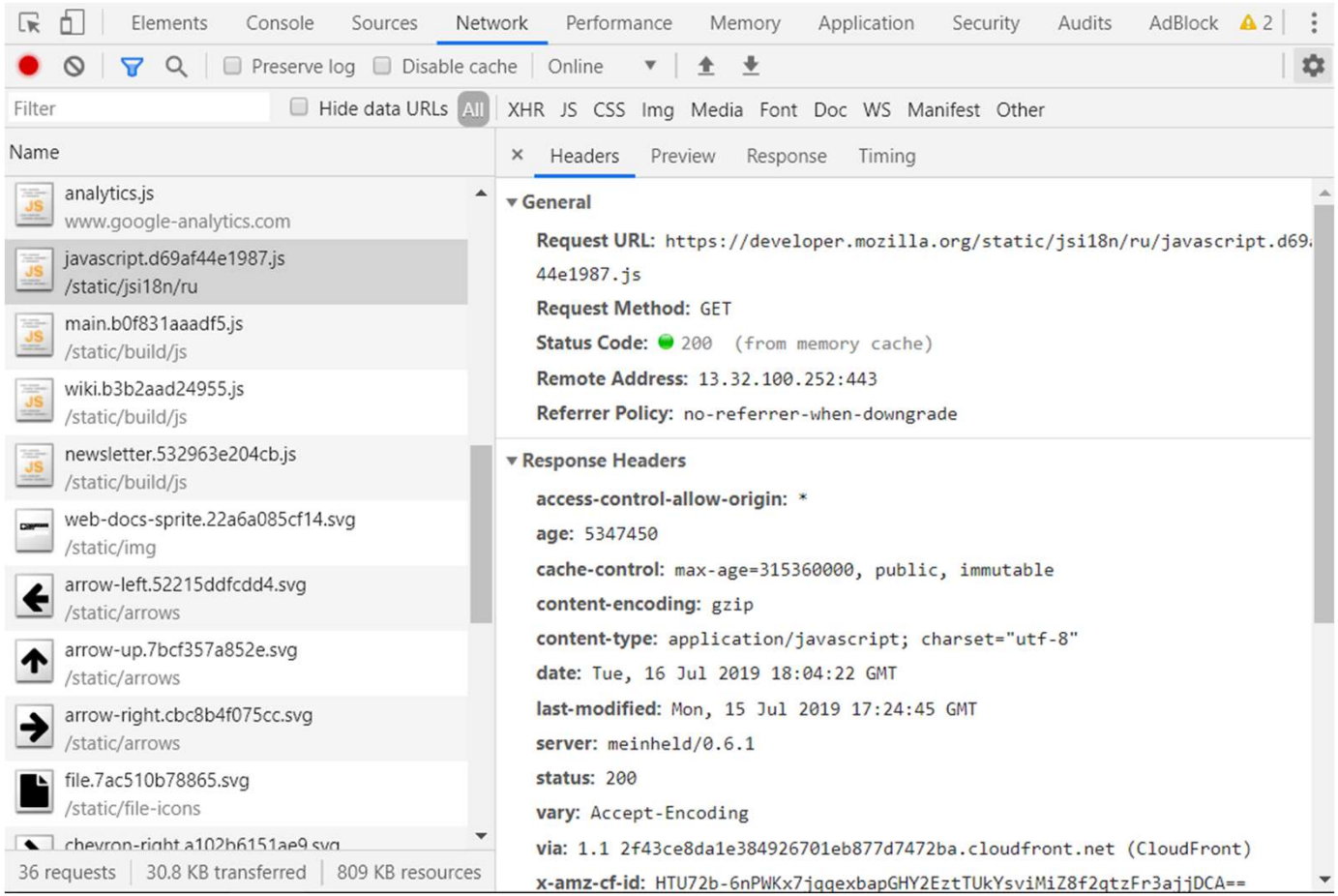
DATA VISUALIZATION



- Timing details visualization
- Load event markers
- Different sorting options:

Start Time, Duration, Latency etc.

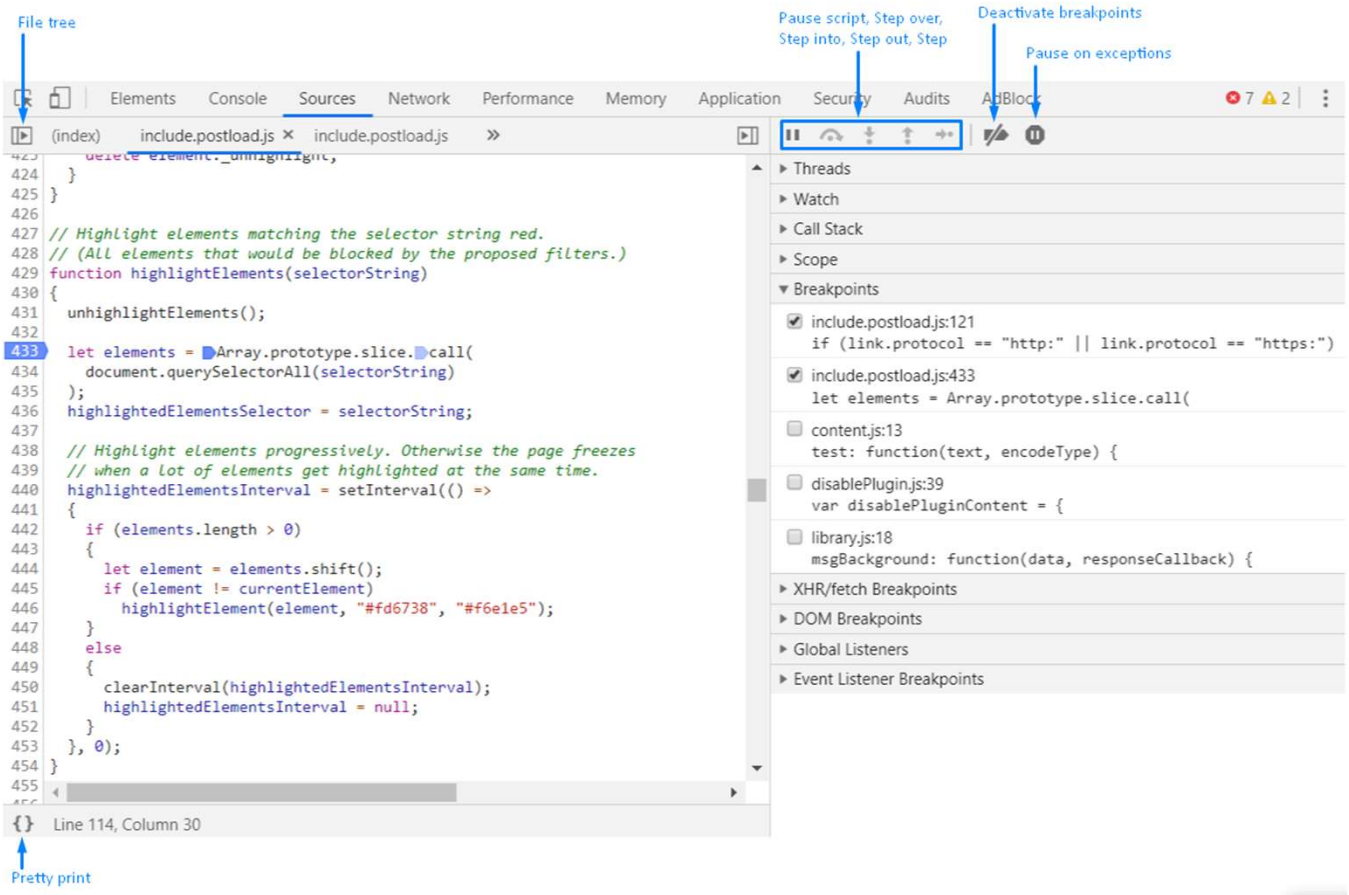
NETWORK RESOURCE DETAILS



- Headers and payload
- Response body
- Cookies
- Timing

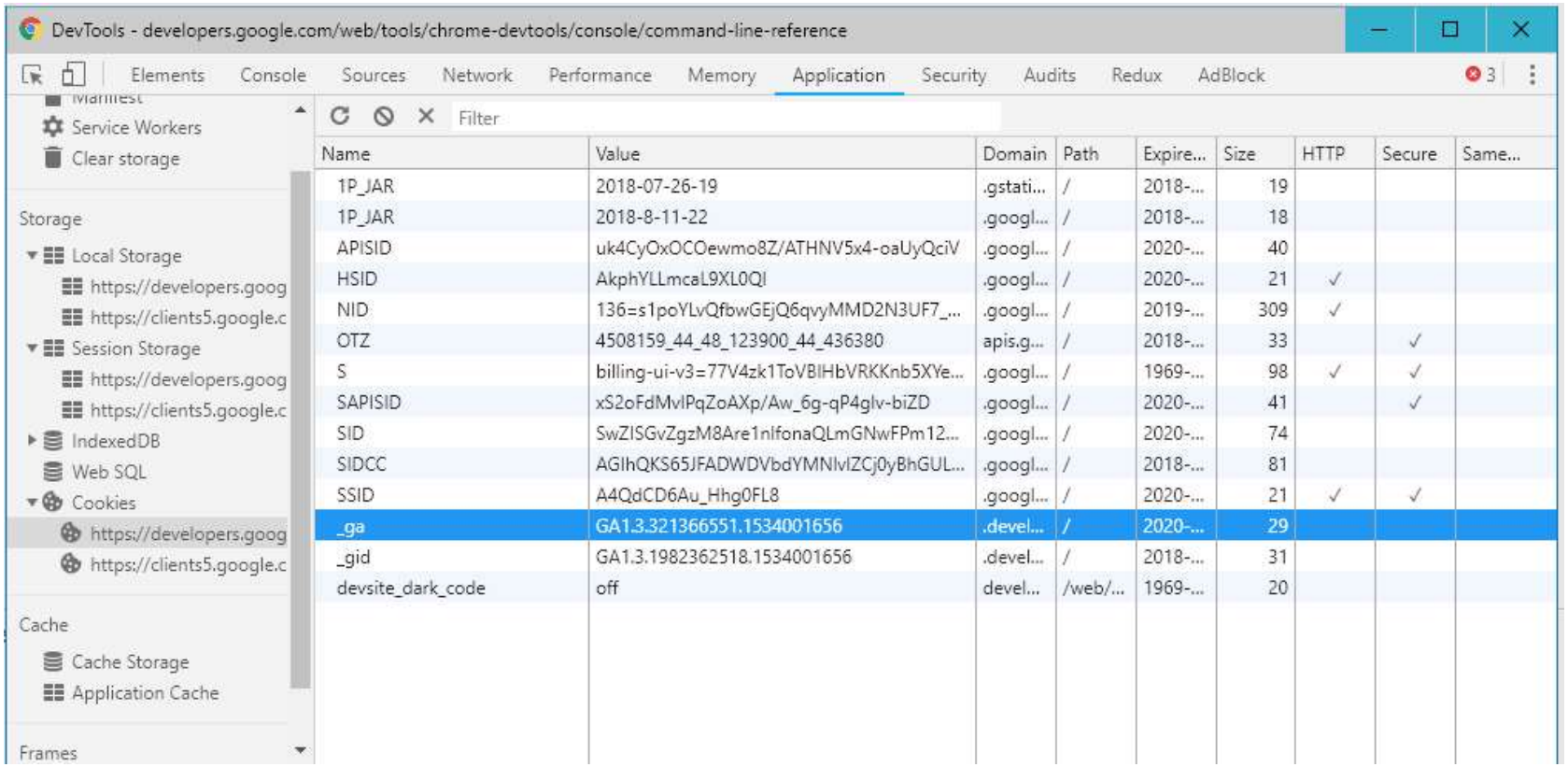
SOURCES

SOURCES PANEL & DEBUGGING JAVASCRIPT



APPLICATION

APPLICATION PANEL



HTML 5 Database

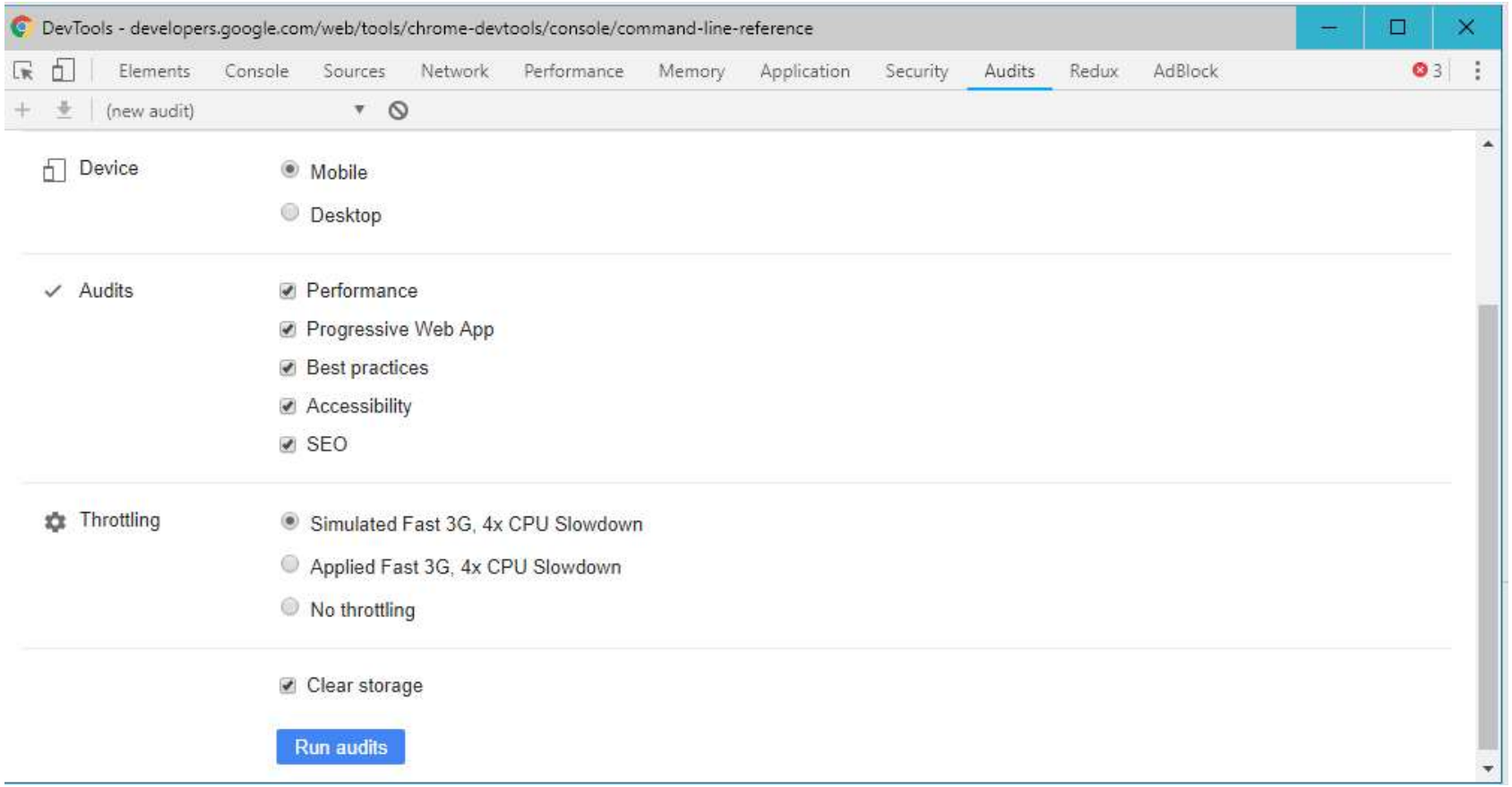
Local Storage

Cookies

AppCache

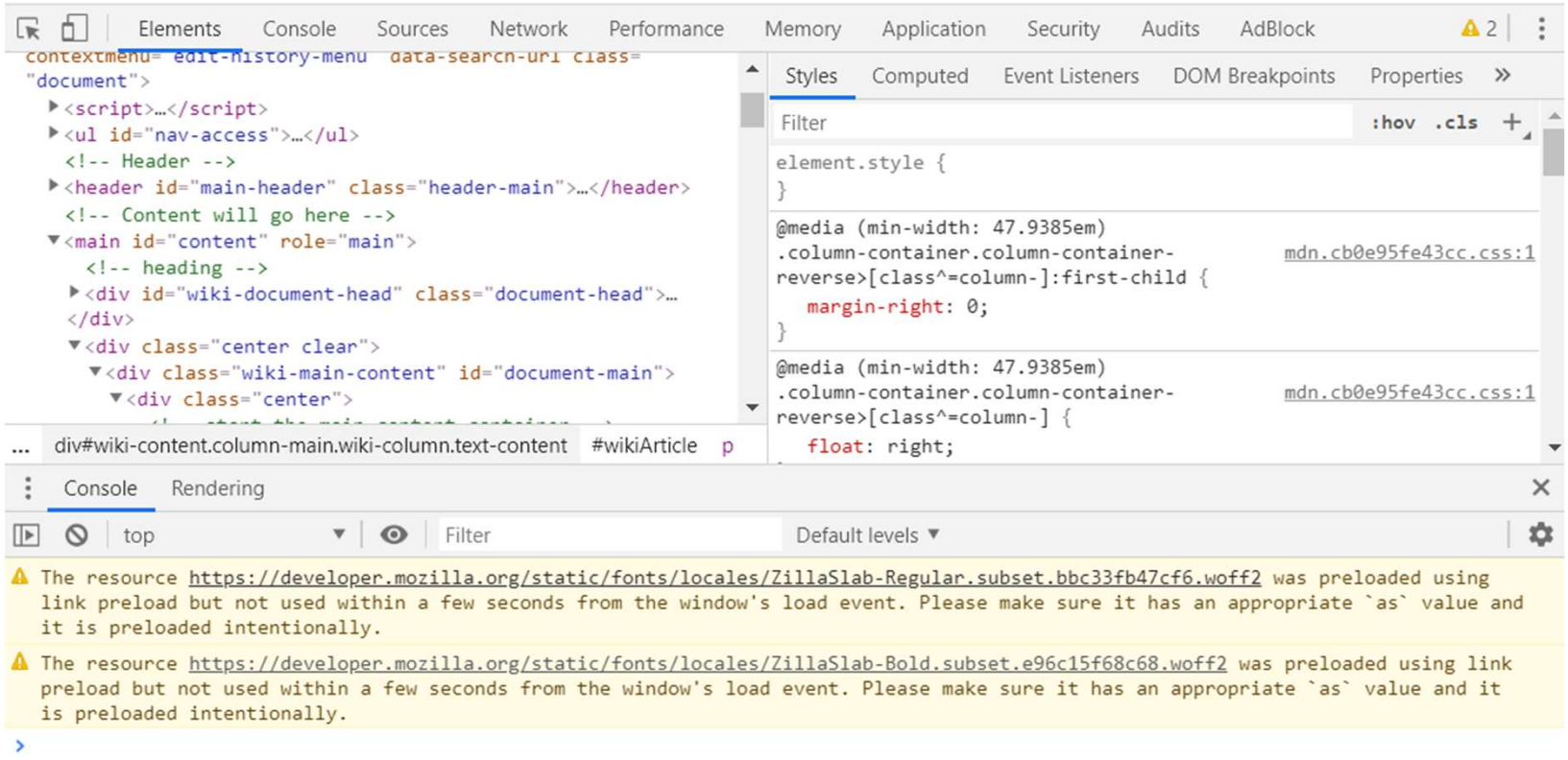
AUDITS

AUDITS PANEL



CONSOLE

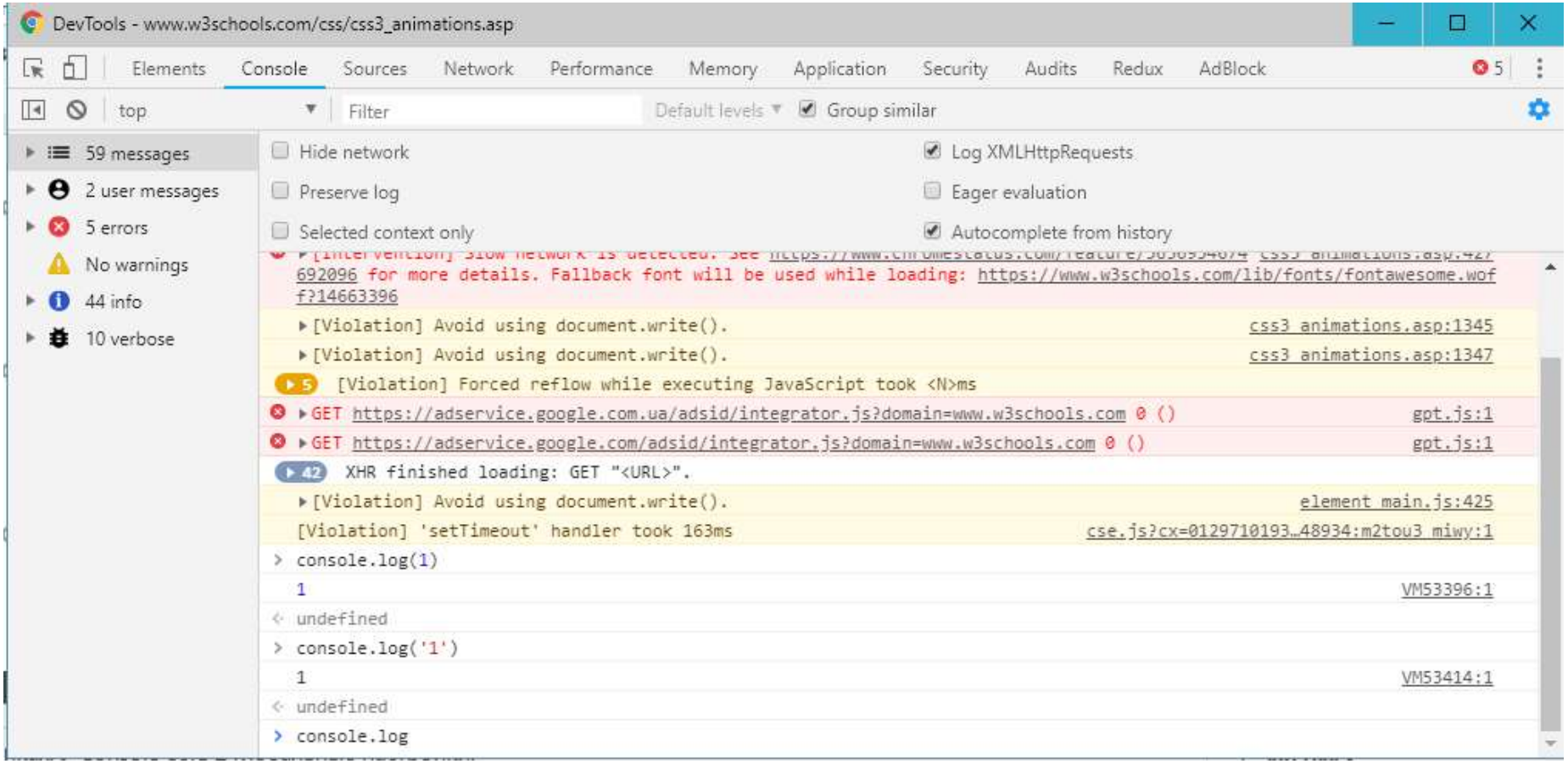
CONSOLE



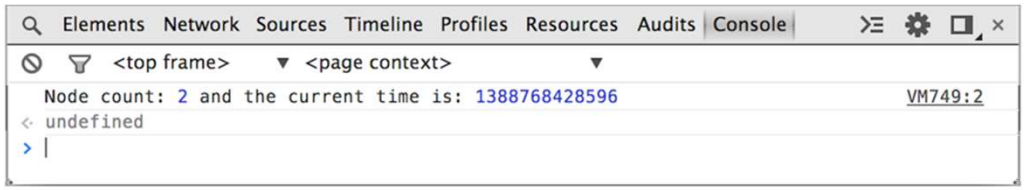
Console API

Command Line API

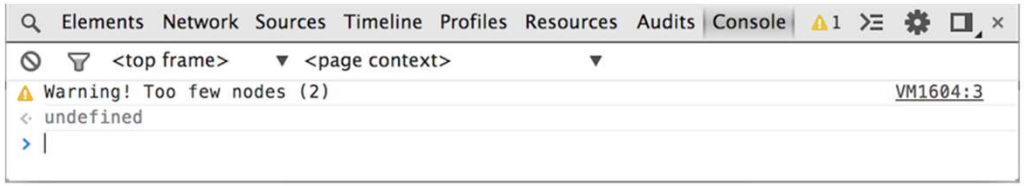
CONSOLE SETTINGS



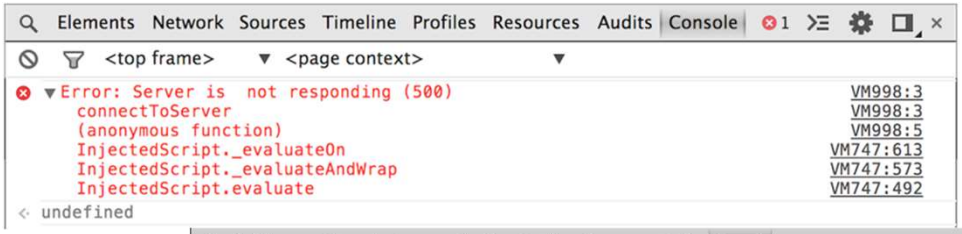
CONSOLE API



console.log()



console.warn()



console.assert()

console.error()

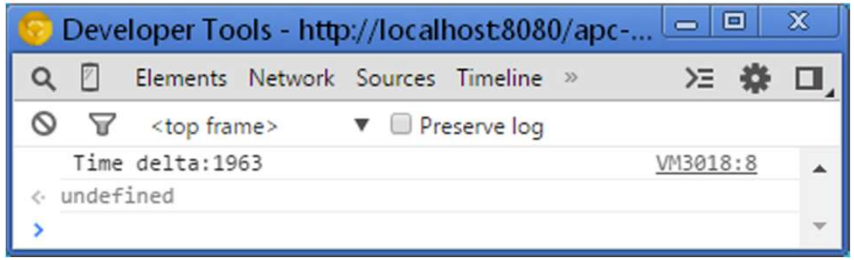
console.trace()

MEASURING HOW LONG SOMETHING TAKES

Old school

JAVASCRIPT

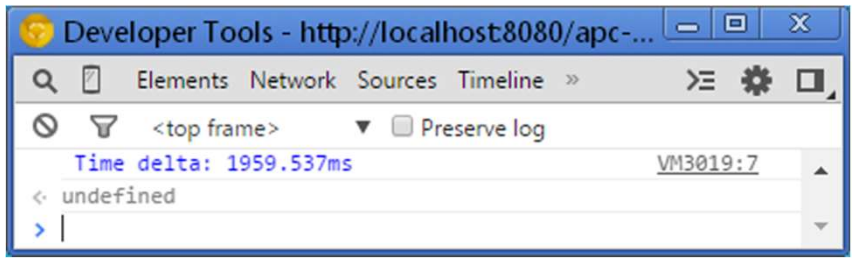
```
var t = new Date();
var array = new Array(1000000);
for(var i=array.length-1; i >=0; i--){
   array[i] = {}
}
t = new Date() - t;
console.log('Time delta:' + t);
```



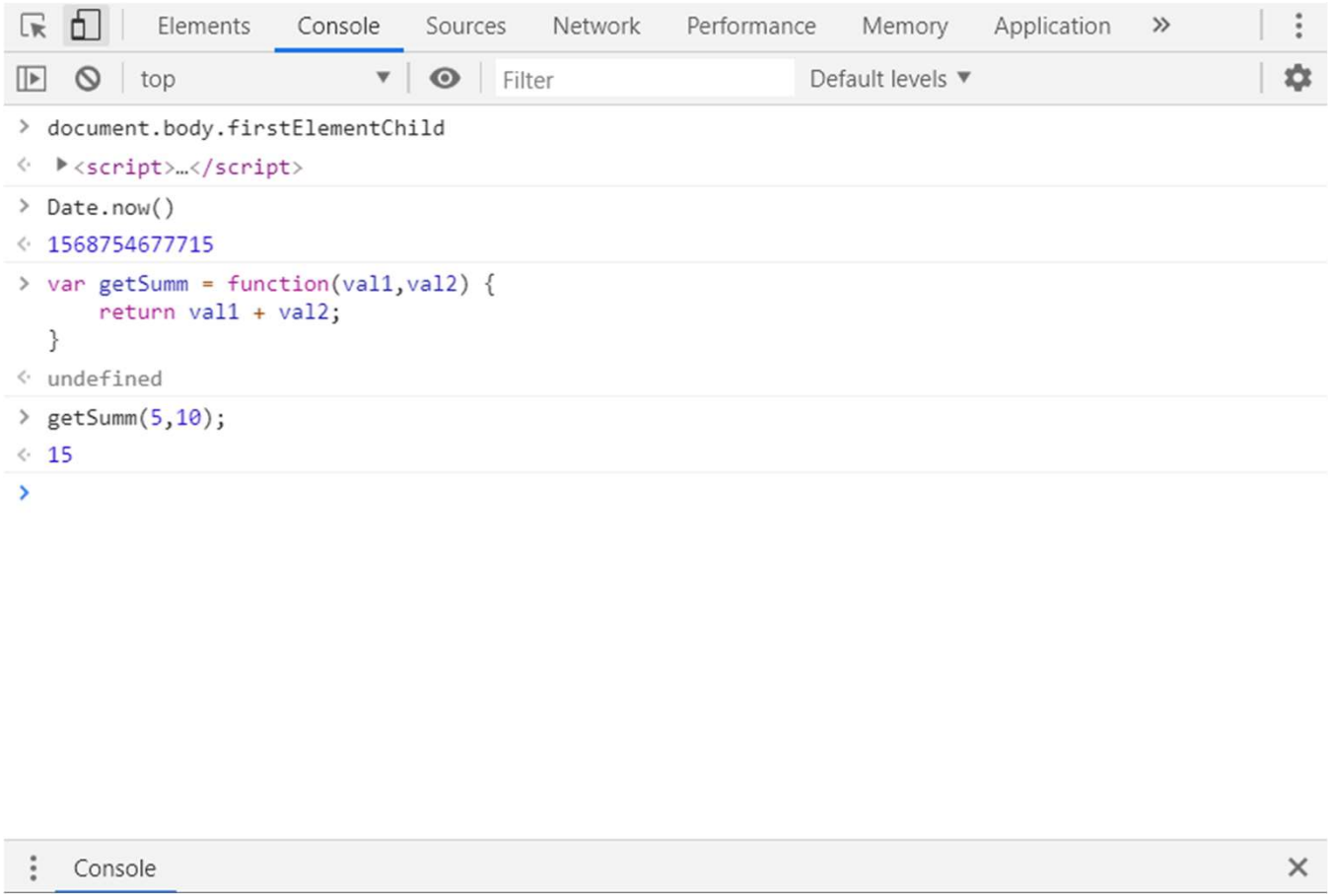
Console API

JAVASCRIPT

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



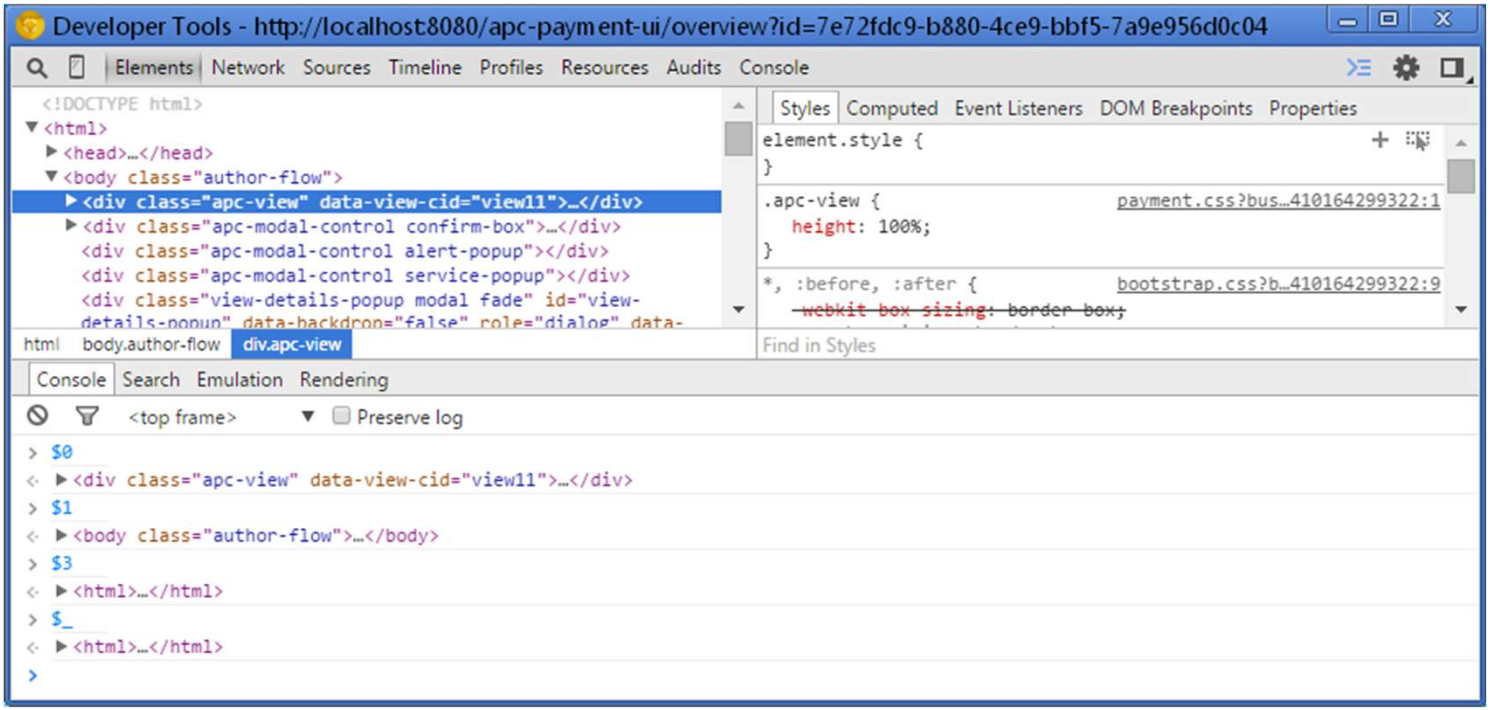
EVALUATING EXPRESSIONS



SELECTING ELEMENTS

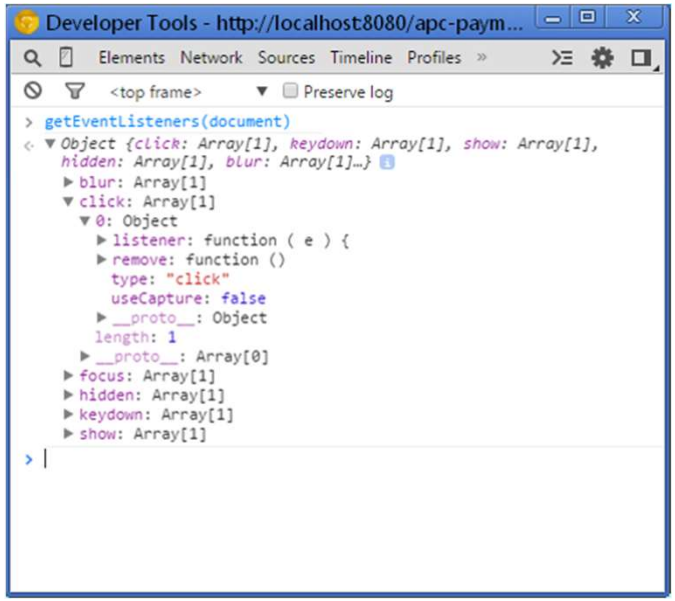
$0, $1, $2, $3, $4
$_

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

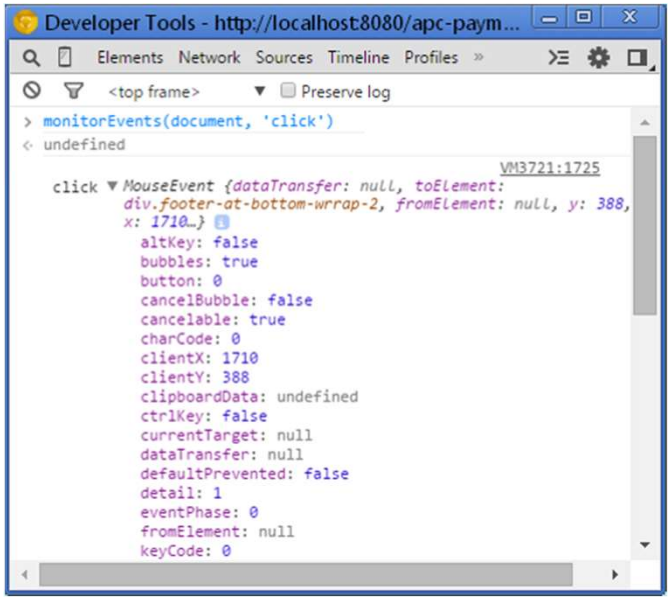


MONITORING EVENTS

getEventListeners(object)



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

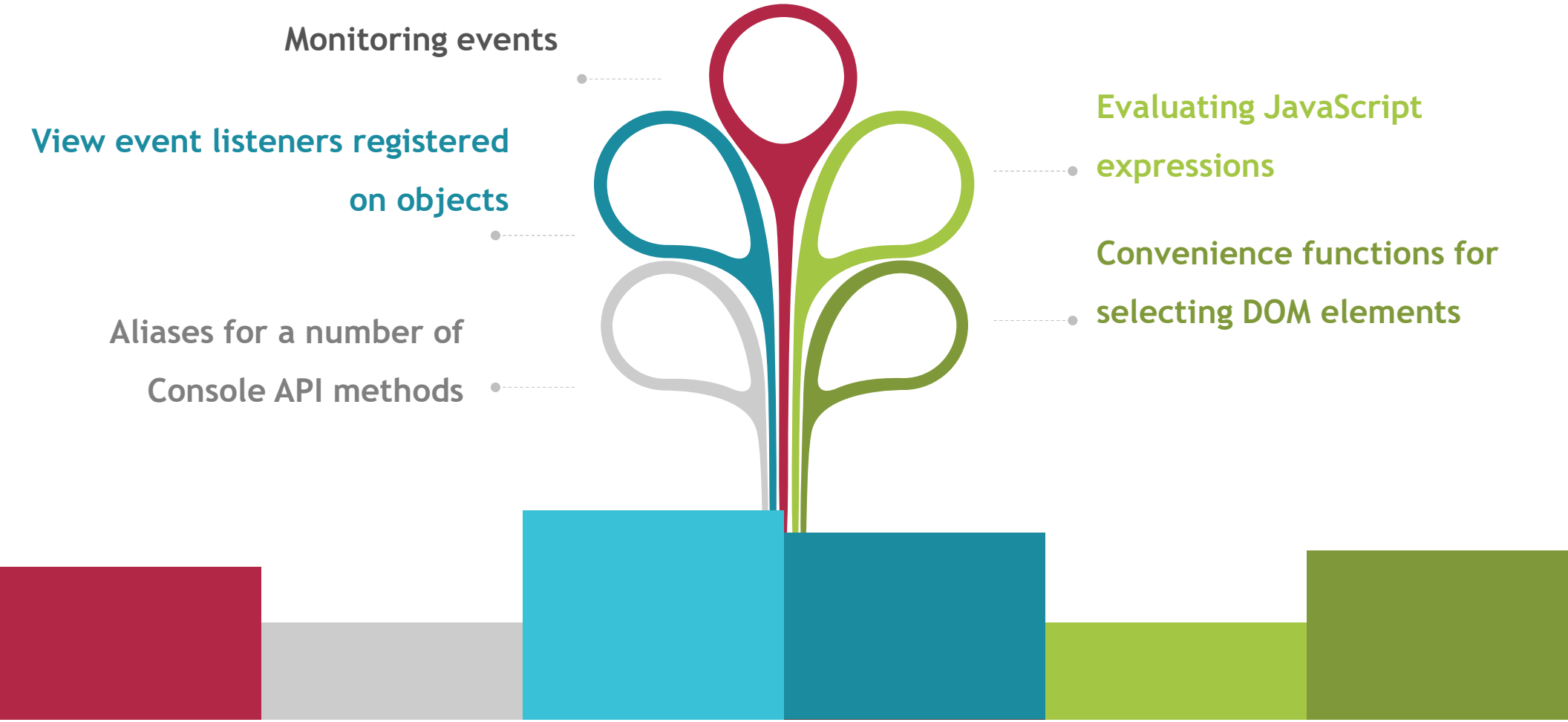


CONSOLE UTILITIES API

Monitoring events

View event listeners registered on objects

Aliases for a number of Console API methods

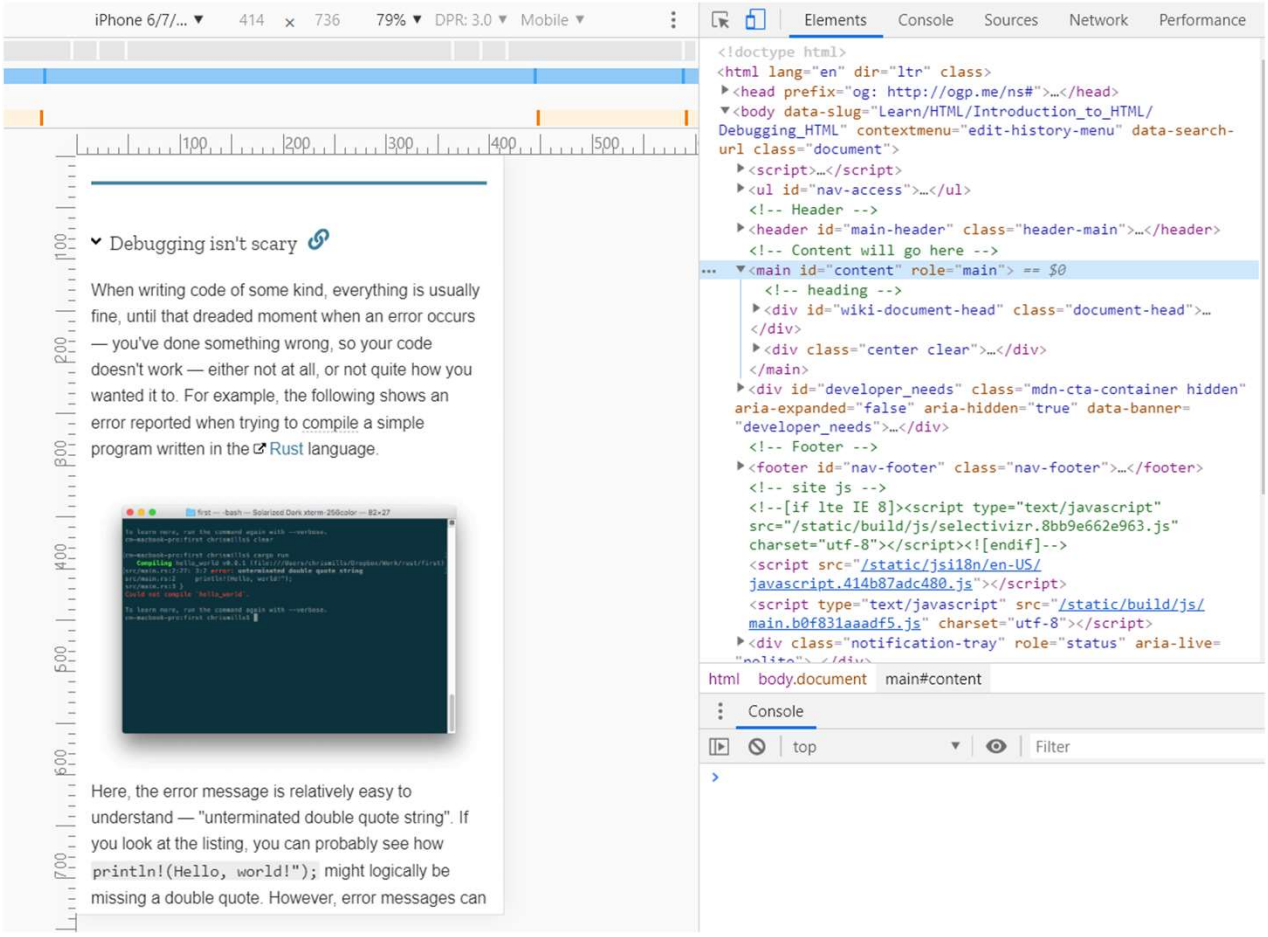


Evaluating JavaScript expressions

Convenience functions for selecting DOM elements

EMULATION

EMULATION

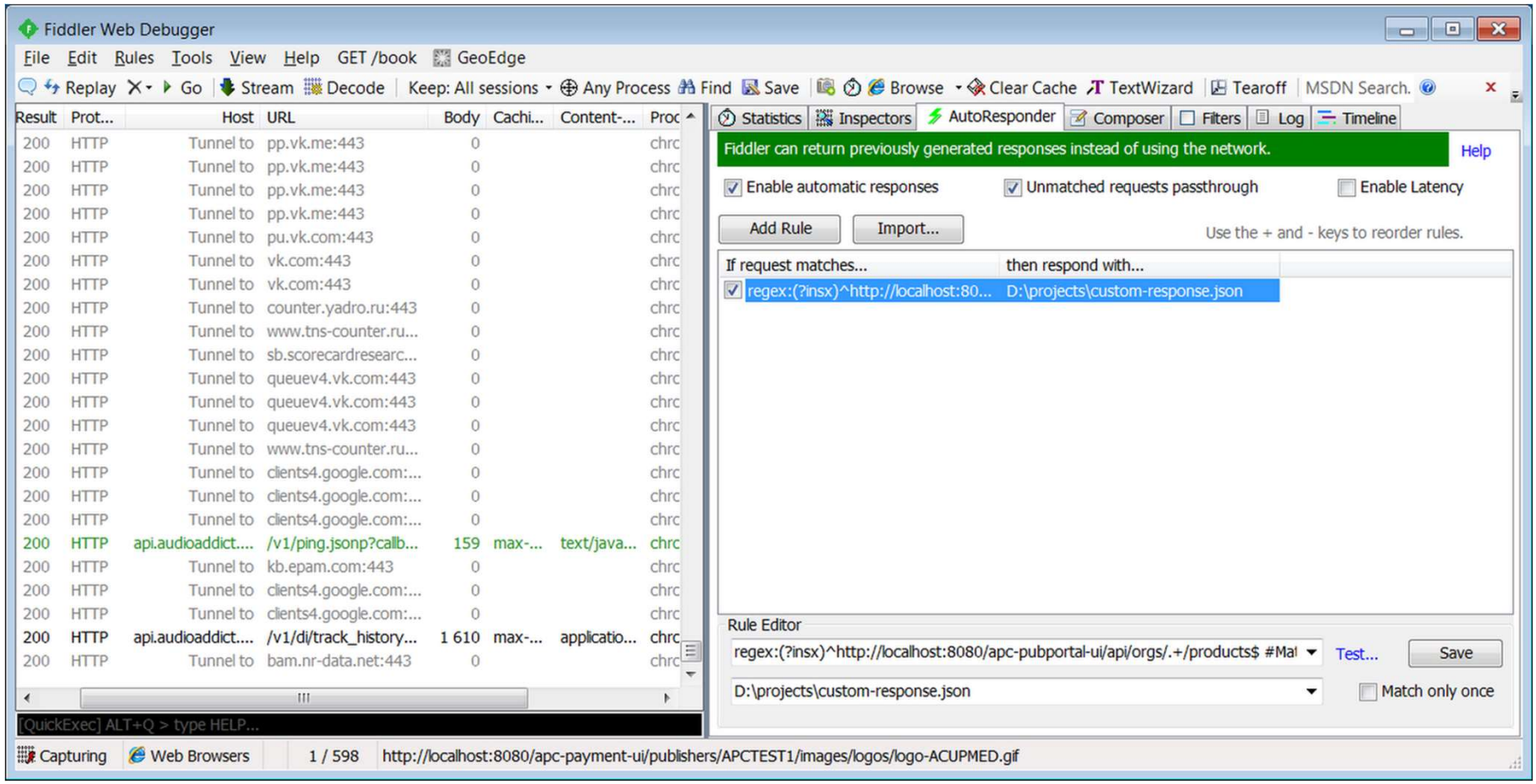

iPhone 6/7/... ▼ 414 × 736 79% ▼ DPR: 3.0 ▼ Mobile ▼
Elements Console Sources Network Performance
Debugging isn't scary
When writing code of some kind, everything is usually fine, until that dreaded moment when an error occurs — you've done something wrong, so your code doesn't work — either not at all, or not quite how you wanted it to. For example, the following shows an error reported when trying to compile a simple program written in the Rust language.
Here, the error message is relatively easy to understand — "unterminated double quote string". If you look at the listing, you can probably see how println!(Hello, world!"); might logically be missing a double quote. However, error messages can
<!doctype html>
<html lang="en" dir="ltr" class>
<head prefix="og: http://ogp.me/ns#">…</head>
<body data-slug="Learn/HTML/Introduction_to_HTML/Debugging_HTML" contextmenu="edit-history-menu" data-search-url class="document">
<script>…</script>
<ul id="nav-access">…</ul>
<!-- Header -->
<header id="main-header" class="header-main">…</header>
<!-- Content will go here -->
<main id="content" role="main"> == $0
<!-- heading -->
…

…
</main>
…
<!-- Footer -->
<footer id="nav-footer" class="nav-footer">…</footer>
<!-- site js -->
<!--[if lte IE 8]><script type="text/javascript" src="/static/build/js/selectivizr.8bb9e662e963.js" charset="utf-8"></script><![endif]-->
<script src="/static/jsi18n/en-US/javascript.414b87adc480.js"></script>
<script type="text/javascript" src="/static/build/js/main.b0f831aaadf5.js" charset="utf-8"></script>
<div class="notification-tray" role="status" aria-live=
html body.document main#content
Console
top
Filter

FIDDLER

FIDDLER

FIDDLER – AUTO RESPONDER

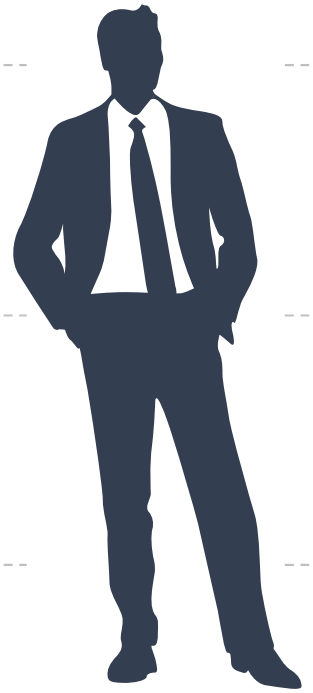


FIDDLER – LIST OF FEATURES

1. We can see all requests, their titles, cookie, parameters transferred to the server.
2. We can test the site on slow connection with the Internet
3. Substitution of data in case of the response of the server

CONCLUSION

DEBUGGING TOOLS



Easy To Use

Intuitively clear interface

Performance

Opportunity to see performance speed

Accuracy

Definitions of problem places and exact description of a problem

See the errors

We can see errors during execution our code

System

Allows to see a system at the time of debugging

Monitoring

We can trace events

RESOURCES

ANY QUESTIONS?

Get in Touch

Vitaliy Malets
Software Engineer
Vitaliy_Malets@epam.com

THANKS FOR ATTENTION!!!