

# DEBUGGING JS

A wide-angle photograph of a turbulent sea under a dramatic sky. The horizon is visible in the distance, where the dark blue of the water meets a sky filled with large, billowing clouds. The clouds are illuminated from behind by a low sun, giving them a fiery orange and red glow. The overall atmosphere is one of raw energy and drama.

# ROADMAP FOR TODAY

» Debugging JS

» I18N

# EXAM EXAMPLE QUESTIONS

- » Write a react component which renders the names of the given users
- » Which performance optimizations can you spot in the following code
- » Name challenges when translating a web application
- » How do you debug performance issues within your application
- » ...

# CONSOLE.LOG DEBUGGING

- » Add console.log statements
- » Click through the app
- » See the log statements in chrome dev tools
  - » easy to get started
  - » some problems might be tough to track down



# USING BREAKPOINTS

- » Chrome/Firefox devtools allow breakpoints
- » Breakpoints are a way to pause a program
- » variables can be inspected
- » custom code can be executed

# USING BREAKPOINTS

- » Chrome/Firefox devtools allow breakpoints
- » Breakpoints are a way to pause a program
- » variables can be inspected
- » custom code can be executed



# HOW TO ADD BREAKPOINTS

- » Via EventListeners
  - » DevTools > Sources > EventListener Breakpoints
- » Via DOM events
  - » eg. element is removed/added to dom
- » In DevTools
  - » Sources > find line of code > click on line number
- » Via Code -> add debugger statement

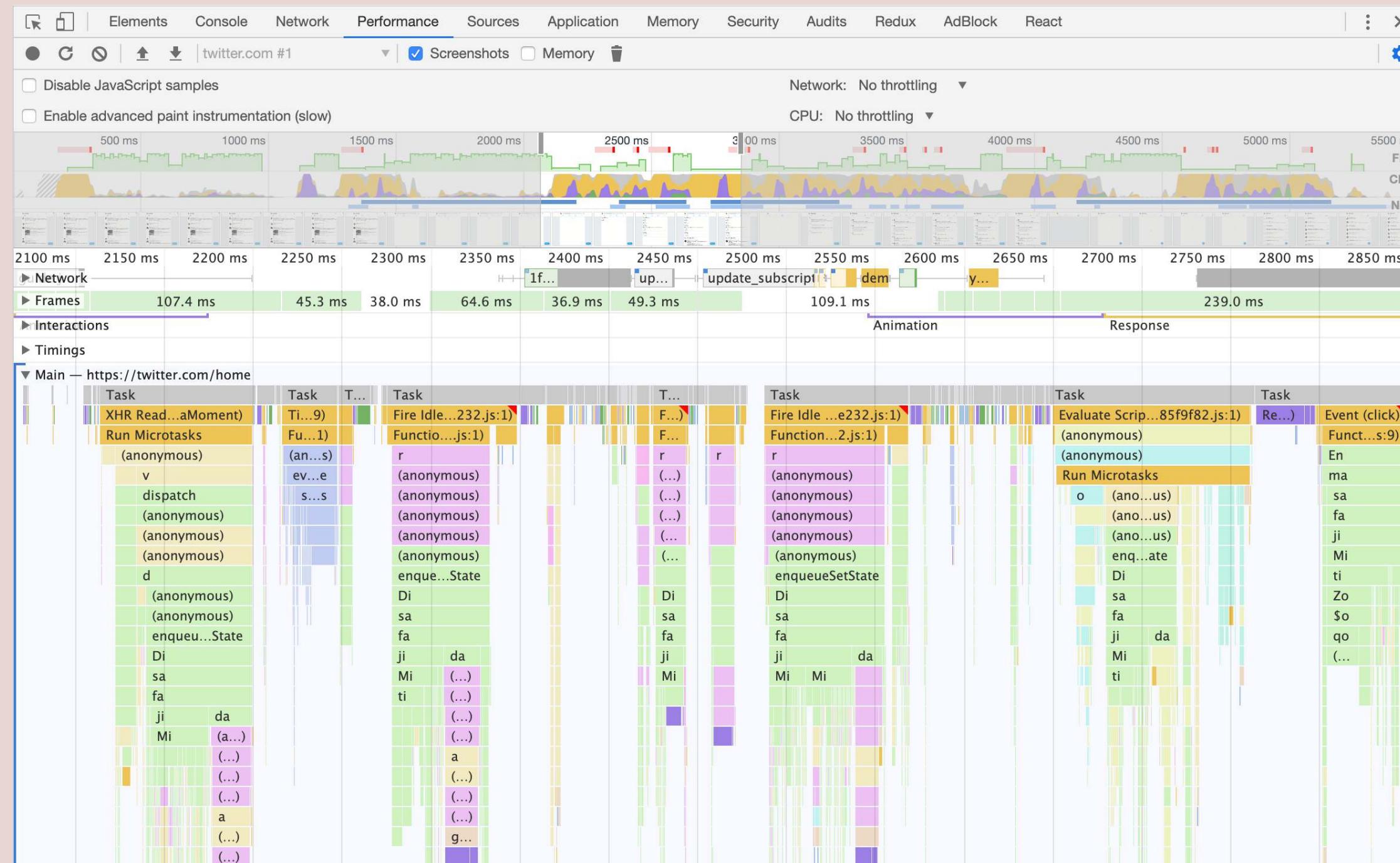
# DEBUGGING NODEJS

» [https://dev.to/john\\_papa/debug-your-nodejs-app-in-60-seconds-5cni](https://dev.to/john_papa/debug-your-nodejs-app-in-60-seconds-5cni)

# DEBUGGING PERFORMANCE ISSUES

A scenic landscape featuring a range of mountains with rugged, light-colored peaks. In the foreground, there are some bare trees and shrubs. A large, semi-transparent red rectangular overlay covers the upper two-thirds of the image, centered on the text.

# CHROME PERFORMANCE TAB



# CHROME PERFORMANCE TAB

- » Determine why CPU is busy
- » Flame graph Visualisation of
  - » call stack
  - » duration of fn call

# BOTTOM UP

» Which functions had took the longest to run

Bottom-Up			
Self Time	Total Time	Activity	
89.4 ms 14.4 %	619.8 ms 100.0 %	Task	
57.4 ms 9.3 %	57.5 ms 9.3 %	▶ Paint	
54.6 ms 8.8 %	54.6 ms 8.8 %	▶ Compile Script	
49.1 ms 7.9 %	61.7 ms 10.0 %	▶ Recalculate Style	
31.9 ms 5.2 %	158.8 ms 25.6 %	▶ Parse HTML	<a href="#">onloadwff.js:71</a>
29.0 ms 4.7 %	29.0 ms 4.7 %	▶ DOM GC	
23.9 ms 3.9 %	94.3 ms 15.2 %	▶ Run Microtasks	
15.1 ms 2.4 %	15.1 ms 2.4 %	▶ Parse Stylesheet	
14.9 ms 2.4 %	118.4 ms 19.1 %	▶ Function Call	
10.5 ms 1.7 %	10.7 ms 1.7 %	▶ LP_looksLikeOTPField	<a href="#">onloadwff.js:71</a>
10.4 ms 1.7 %	10.4 ms 1.7 %	▶ Layout	
7.8 ms 1.3 %	7.8 ms 1.3 %	▶ Update Layer Tree	
6.8 ms 1.1 %	8.9 ms 1.4 %	▶ isGlobalProperty	<a href="#">onloadwff.js:12</a>
6.7 ms 1.1 %	6.7 ms 1.1 %	▶ querySelectorAll	
6.3 ms 1.0 %	8.2 ms 1.3 %	▶ n	
5.9 ms 1.0 %	5.9 ms 1.0 %	▶ Minor GC	

# CPU PERFORMANCE

- » Open dev tools
- » Go to Performance > Record
- » Click stop

# TASK

- » clone: <https://github.com/webpapaya/compup>
- » npm i && npm run start:development
- » Users complain that some button clicks take very long
- » Don't look on commit history =)

# FEEDBACK

- » Questions: [tmayrhofer.lba@fh-salzburg.ac.at](mailto:tmayrhofer.lba@fh-salzburg.ac.at)
- » <https://s.surveyplanet.com/xlibwm85>