front-end

Build && Performance

lab 5/8

today

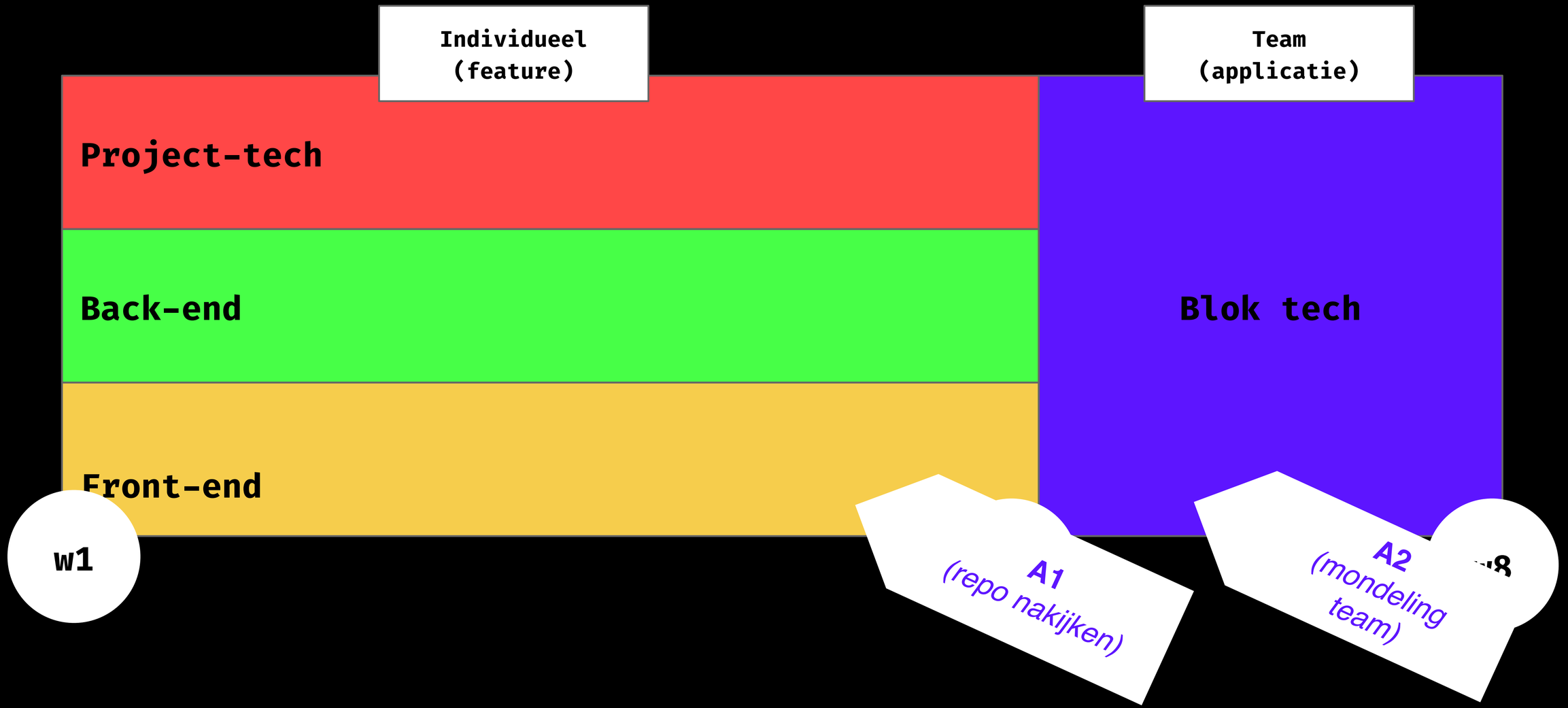
I. ~~Standup~~

II. Team assignment (A2)

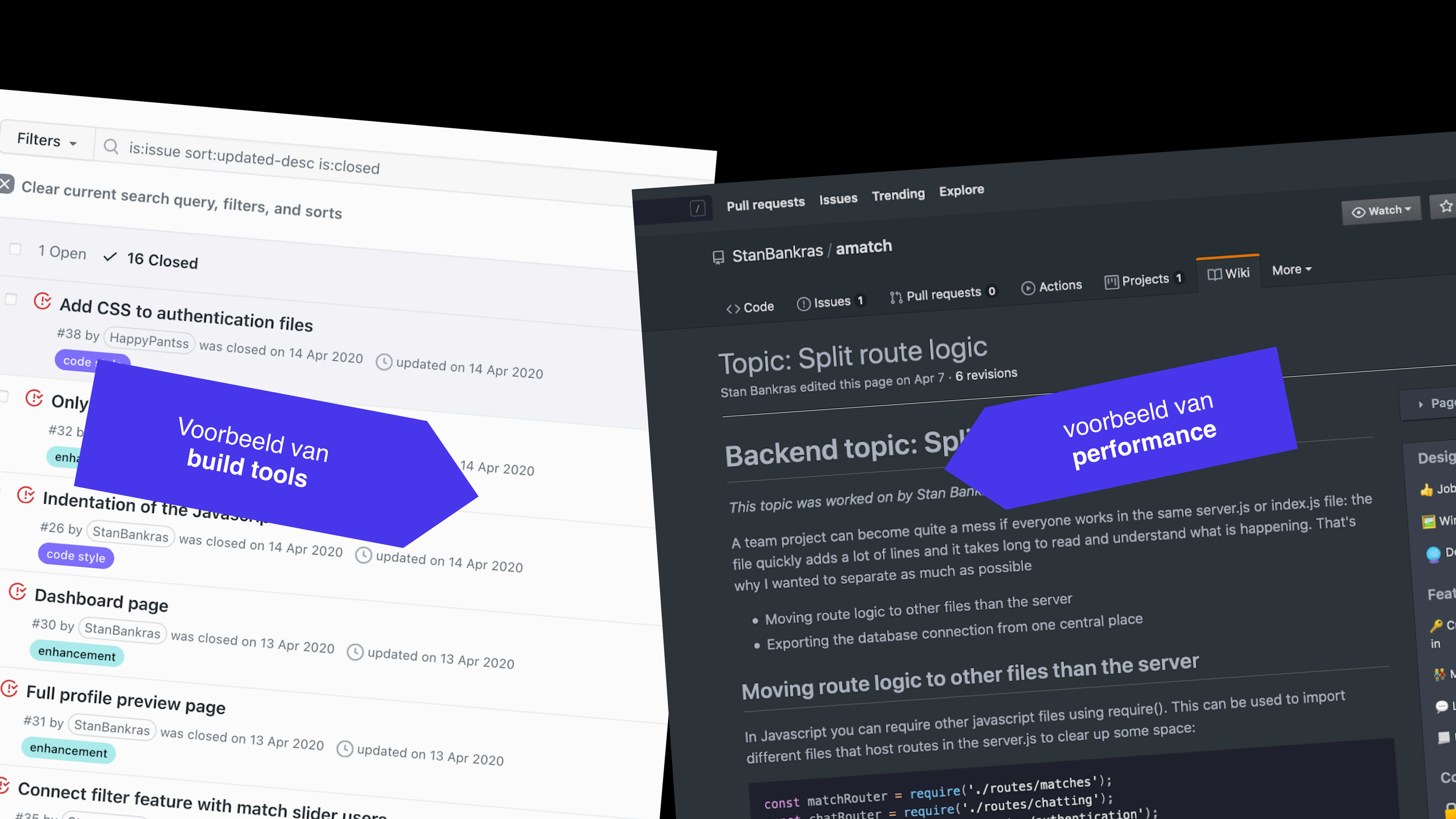
III. Build Tools

IV. Performance

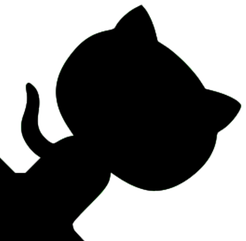
Team assignment


Individueel
(feature)
Team
(applicatie)
Project-tech
Back-end
Front-end
Blok tech
w1
A1
(repo nakijken)
A2
(mondeling
team)

	1-2	3-4	5-6	7-8	9-10
Application	There is no production optimization	The production optimization isn't implemented in code or partially implemented but gives errors and is buggy	There is a production optimization and the technical implementation works. The set-up is integrated into the project and merged.	Ther… op… impl… works a… members benefitted of code you've written	…production … resilient by streamlining the asset pipeline.
Research	There is no research in the wiki	There is research in the wiki but very limited, not enough to understand your process	There is research about what production topic you've implemented in the team repo, you've documented your process and ideas and you've researched your topic. You've written about how you implemented your production topic.	The res… in-d… des… you w… code and did research on alternatives	The wiki reads like a … Your production … progressive enhancement and code.
Quality	There is no cohesion in the front-end code of the application. All features are individually made and don't	There was an attempt to unify the front-end in a single application, but it's buggy or not	Front-end architecture has been normalised among all team members. There are basic guidelines each team member follows. The HTML, CSS and JavaScript are written using previous standard.	The application is r… The strateg… and code is modular and applies only	…plication has … professional development agency.


Filters
is:issue sort:updated-desc is:closed
Clear current search query, filters, and sorts
1 Open
16 Closed
Add CSS to authentication files
#38 by HappyPantss was closed on 14 Apr 2020
updated on 14 Apr 2020
code style
Only
#32 b
enha
14 Apr 2020
Indentation of the Javascr
#26 by StanBankras was closed on 14 Apr 2020
updated on 14 Apr 2020
code style
Dashboard page
#30 by StanBankras was closed on 13 Apr 2020
updated on 13 Apr 2020
enhancement
Full profile preview page
#31 by StanBankras was closed on 13 Apr 2020
updated on 13 Apr 2020
enhancement
Connect filter feature with match slider users
Voorbeeld van
build tools
Pull requests
Issues
Trending
Explore
Watch
StanBankras / amatch
Code
Issues 1
Pull requests 0
Actions
Projects 1
Wiki
More
Topic: Split route logic
Stan Bankras edited this page on Apr 7 · 6 revisions
Backend topic: Spl
This topic was worked on by Stan Bank
A team project can become quite a mess if everyone works in the same server.js or index.js file: the file quickly adds a lot of lines and it takes long to read and understand what is happening. That's why I wanted to separate as much as possible
Moving route logic to other files than the server
Exporting the database connection from one central place
Moving route logic to other files than the server
In Javascript you can require other javascript files using require(). This can be used to import different files that host routes in the server.js to clear up some space:
const matchRouter = require('./routes/matches');
const chatRouter = require('./routes/chatting');
voorbeeld van
performance

Assignment (45m)



Think about a topic (production optimisation) you want to work on for front-end.

Tip; check resources on FeD repo or have a look at example repo’s from previous groups.

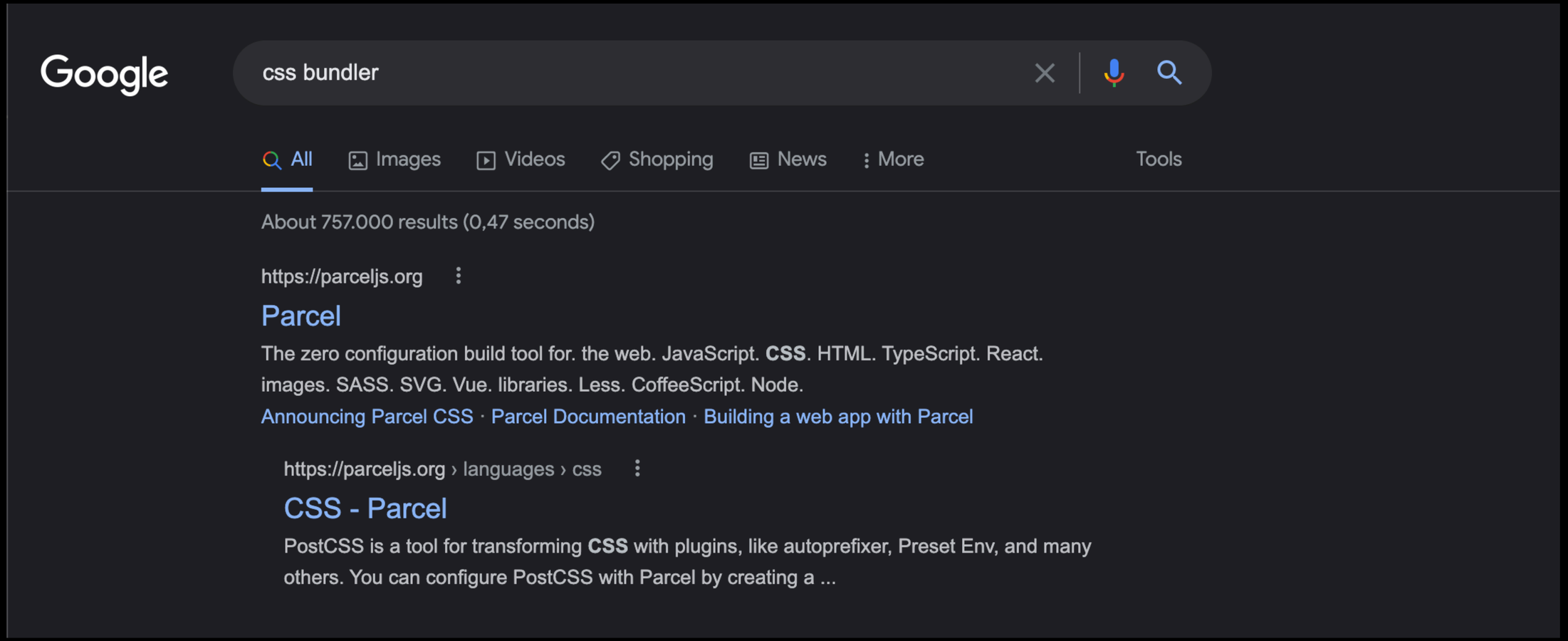
Build Tools

Components

CSS

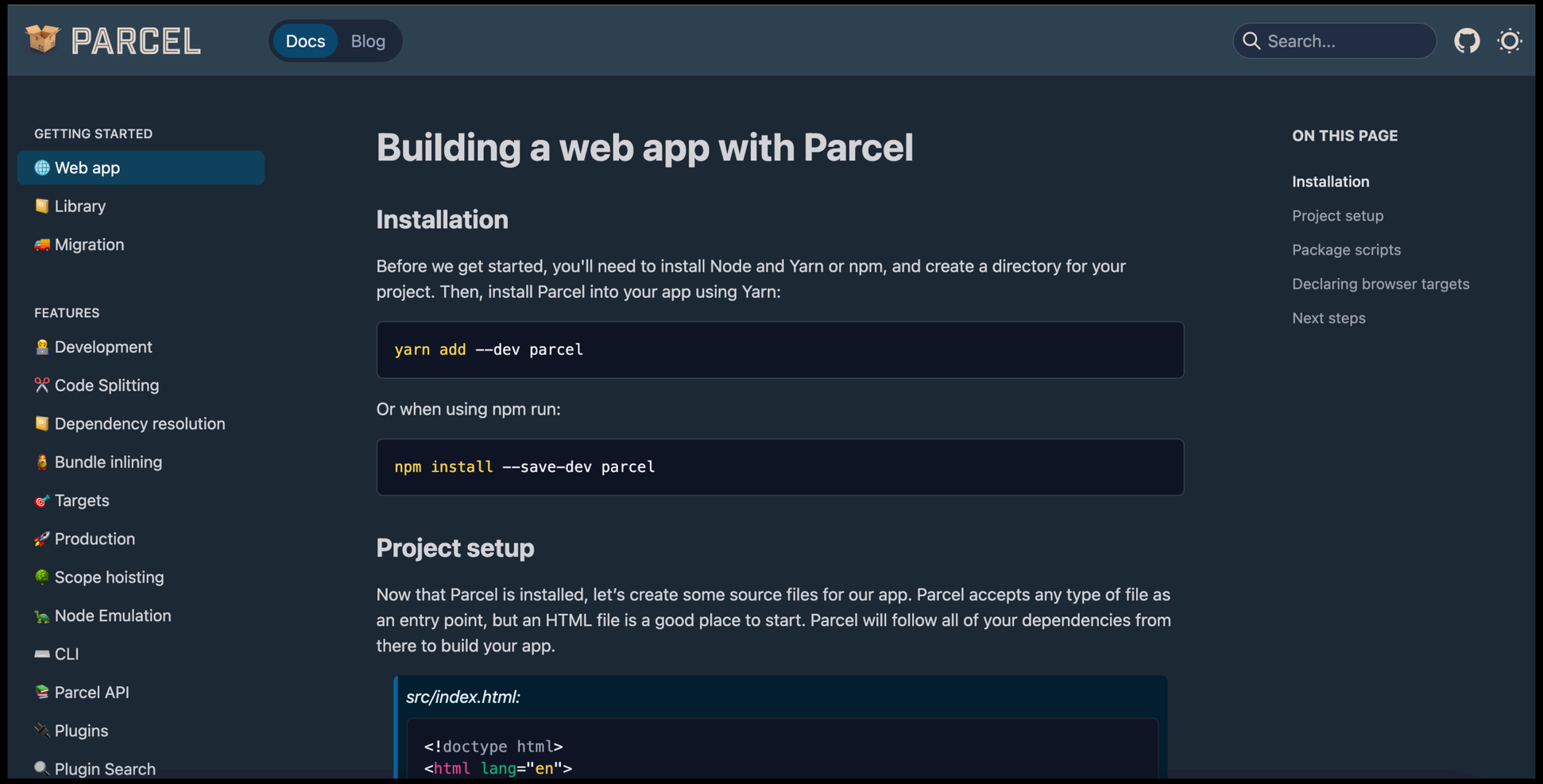
Components

Bundler



Components

Bundler

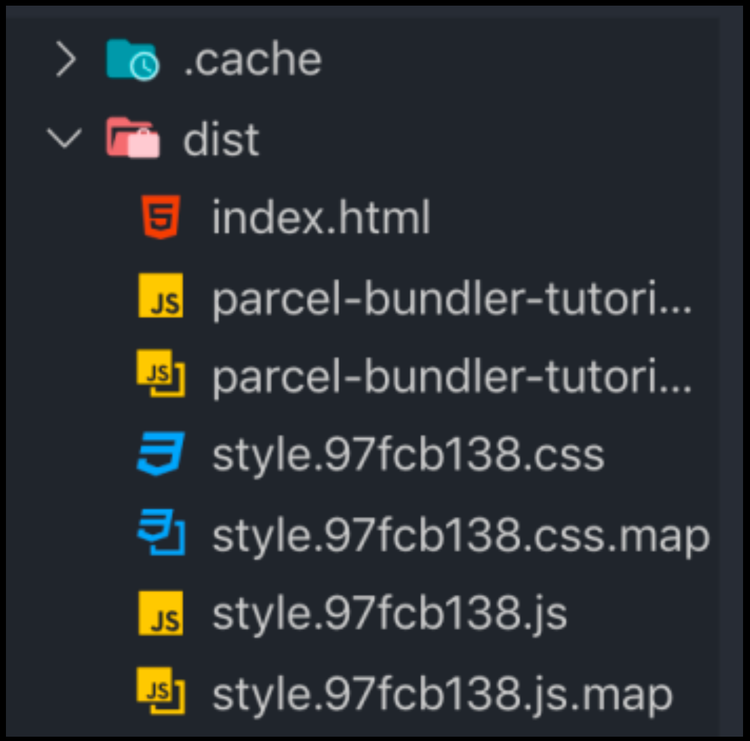


Components

Bundler

```
✨ Built in 444ms.

dist/parcel-bundler-tutorial.8bdf8f45.js      1.48 KB    241ms
dist/style.134bbd53.css.map                     779 B      5ms
dist/parcel-bundler-tutorial.8bdf8f45.js.map    756 B      2ms
dist/index.html                                 639 B      5ms
dist/style.134bbd53.css                         337 B     13ms
```



Components

Bundler

Gulp GET STARTED API PLUGINS DONATE ENTERPRISE Search

TypeScript
Develop in any language

PNG
Create assets with any tool

Markdown
Write using any format

A toolkit to automate & enhance your workflow

Leverage gulp and the fl
slow, repetitive workflow
build pipelines.

rollup.js

- Introduction
  - Overview
  - Installation
  - Quick start
  - The Why
  - Tree-Shaking
  - Compatibility
- Command Line Interface
  - Configuration Files
  - Differences to the JavaScript API
  - Loading a configuration from a Node package
  - Using untranspiled config files
  - Command line flags
  - Reading a file from stdin
- JavaScript API
  - rollup.rollup
  - rollup.watch

rollup.js

Introduction

- Overview
- Installation
- Quick start
- The Why
- Tree-Shaking
- Compatibility
  - Importing CommonJS
  - Publishing ES Modules

Black Lives Matter

Getting Started Configuring Tasks Plugins

GRUNT

The JavaScript Task Runne

atest Version

- Stable: v1.4.1 (npm)
- Development: HEAD (GitHub)

Ads by Bocoup.

Latest News

Grunt 1.4.0 released

Why use a task runner?

In one word: automation. The less work you have to do when performing repetitive tasks like minification, compilation, unit testing, linting, etc, the easier your job becomes. After you've configured it through a Gruntfile, a task runner can do most of that mundane work for you—and your team—with basically zero effort.

Why use Grun

The Grunt ecosystem is hu
every day. With literally hu
choose from, you can use
just about anything with a
If someone hasn't already
need, authoring and publis
Grunt plugin to npm is a b
get started.

Build tools

Ideas

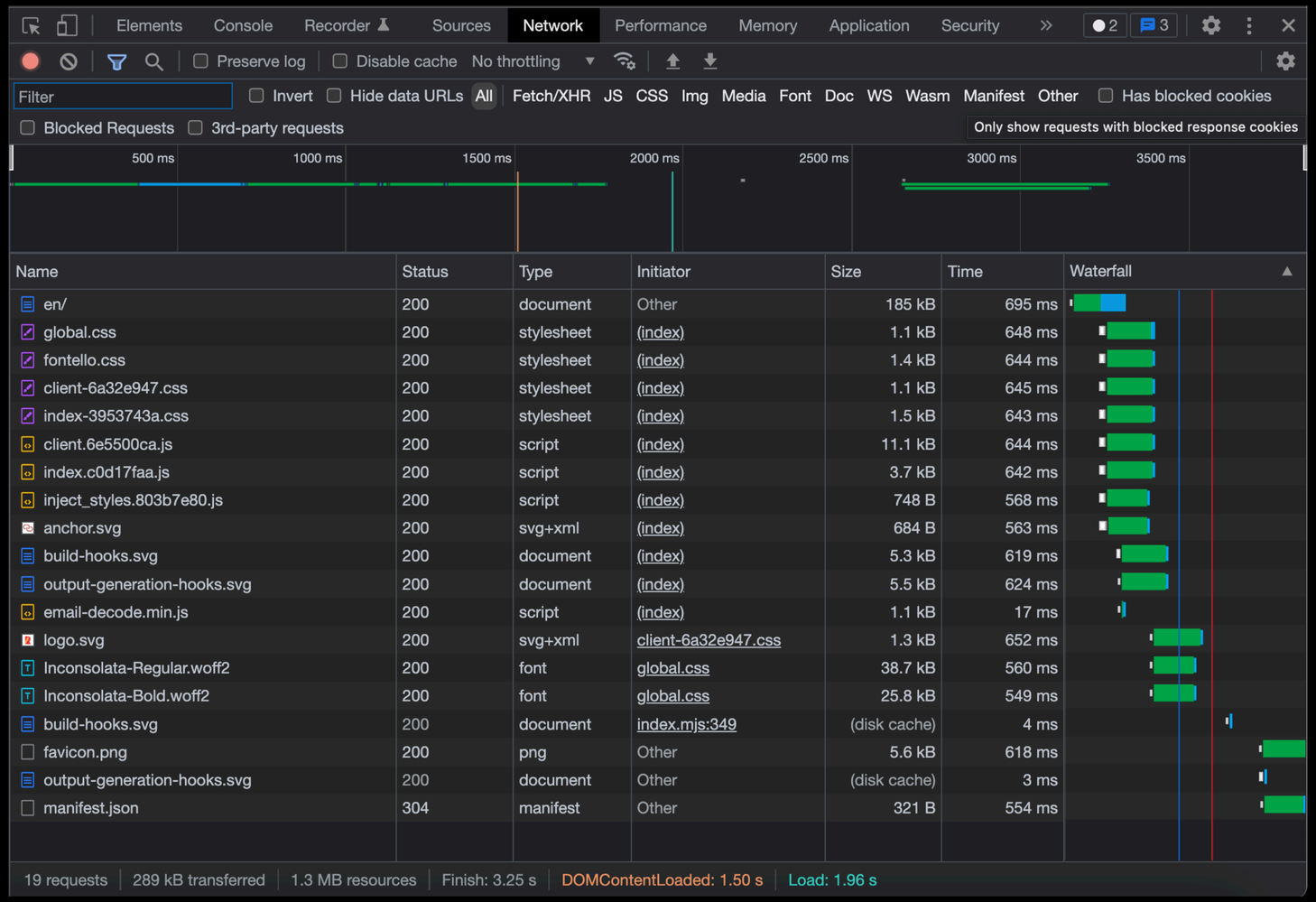
- Bundle CSS (post/pre processor)
- Bundle JS (bundler, npm scripts)
- Use a task runner (gulp, grunt)

Break!

Performance

Performance

What ?



Performance

What ?

Web performance refers to how quickly site content **loads and renders** in a web browser, and how well it responds to user interaction. Bad performing sites are **slow to display and slow to respond to input.** Bad performing sites increase site abandonment.

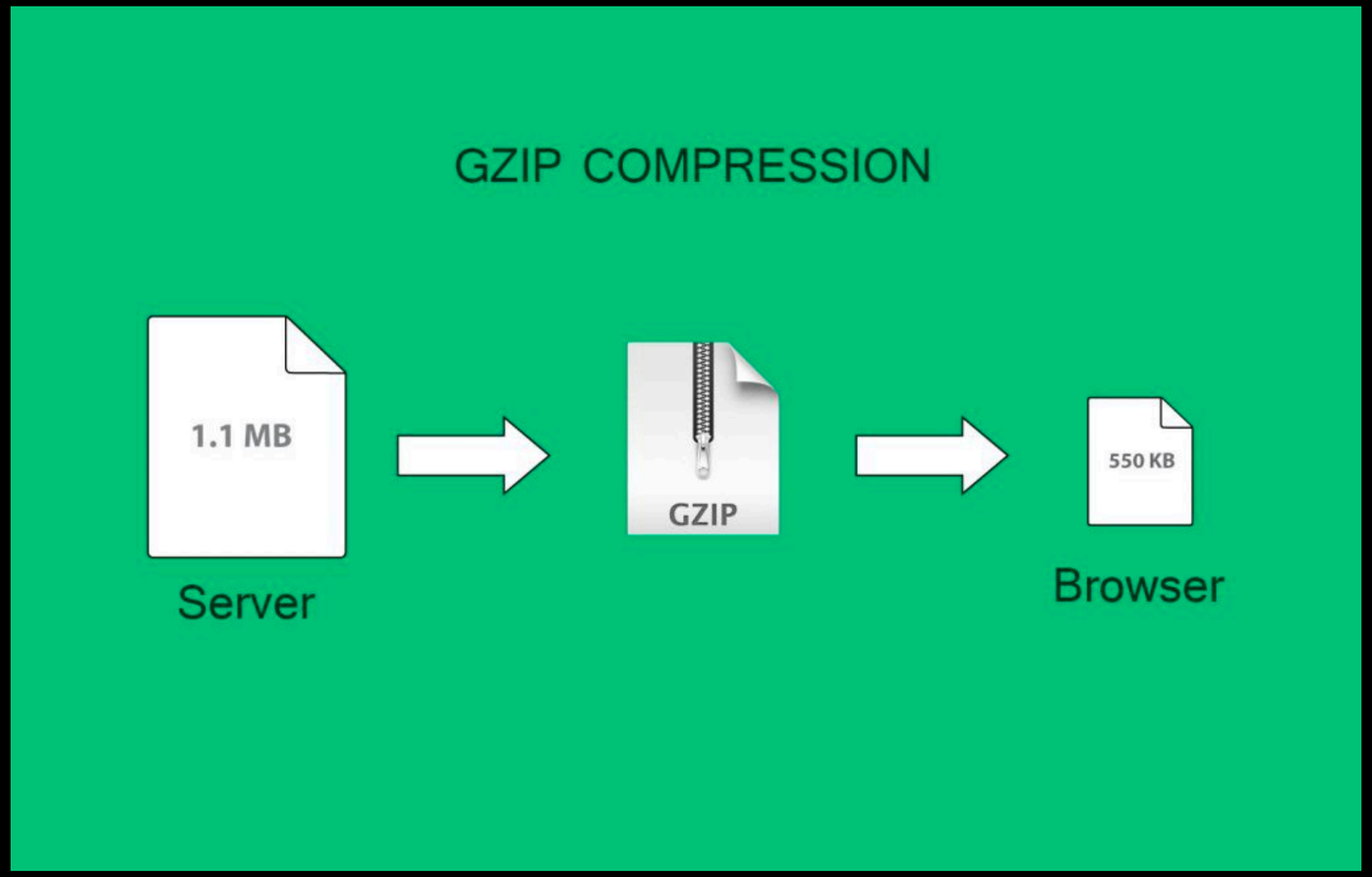
Performance

What?

- Minimize assets (img / css / js / html)
- Minify requests (img / css / js / ???)
- Remove “render-blocking” resources
- Use caching appropriately
- Compress assets (gzip)

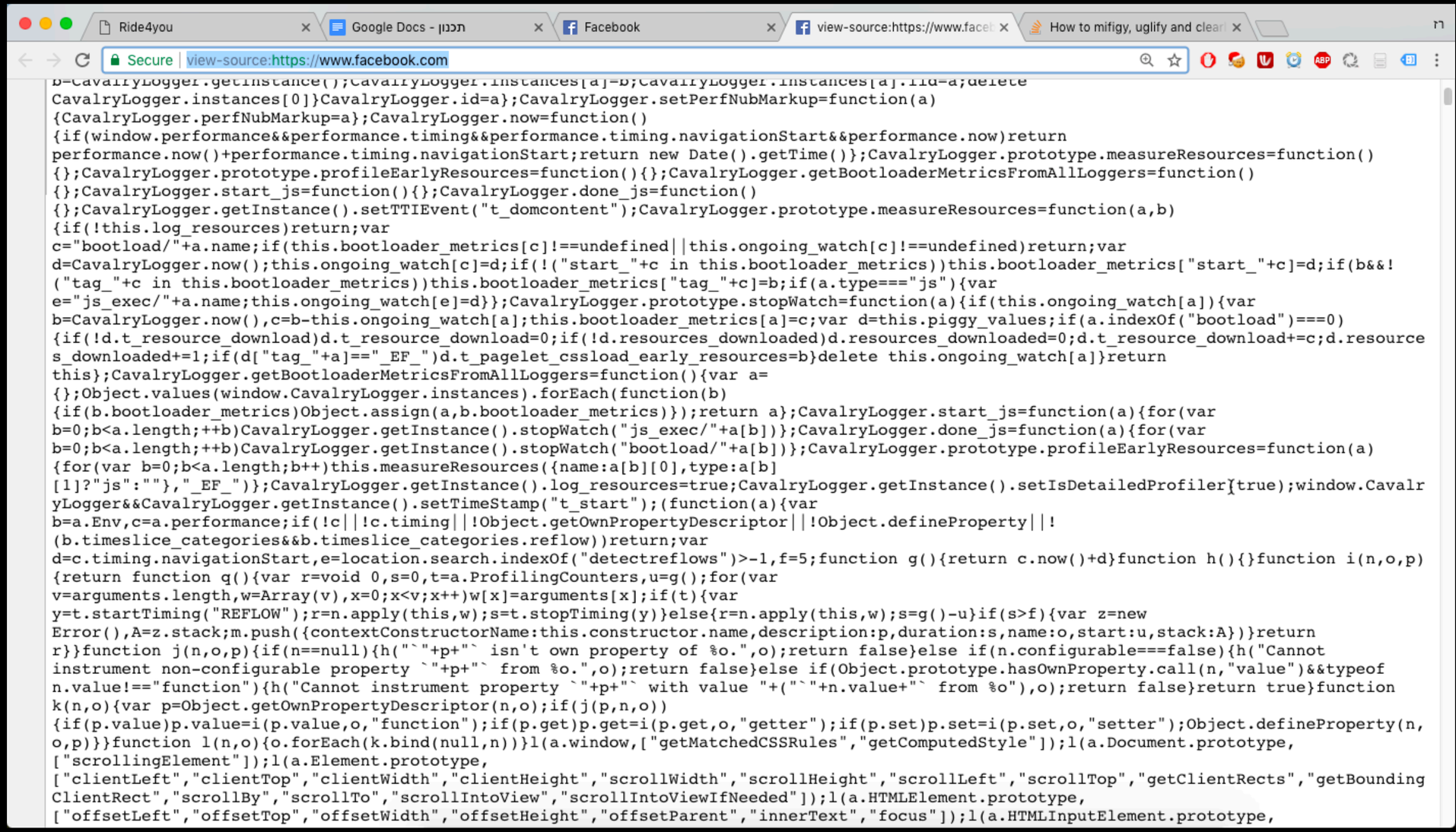
Performance

What?



Performance

Gzip



Performance

Lazyloading

The `loading` attribute

Today, Chrome already loads images at different priorities depending on where they're located with respect to the device viewport. Images below the viewport are loaded with a lower priority, but they're still fetched as soon as possible.

In Chrome 76+, you can use the `loading` attribute to completely defer the loading of offscreen images that can be reached by scrolling:

```
<img src="image.png" loading="lazy" alt="…" width="200" height="200">
```

Here are the supported values for the `loading` attribute:

- `auto`: Default lazy-loading behavior of the browser, which is the same as not including the attribute.
- `lazy`: Defer loading of the resource until it reaches a calculated distance from the viewport.
- `eager`: Load the resource immediately, regardless of where it's located on the page.

Assignment (30 min)

Look into performance / optimisation. Where could your application benefit from it? Can you optimise images on uploading? Lazyload other images? Lazyload fonts? Minify HTML/CSS/JavaScript? Write down your research and include it in your wiki, and if possible, include it in your application.

Demo

exit;

see you in **next Q&A session.**