

WEBPACK 5

WHY BREAKING CHANGES?

A CHANGELOG ONLY TELLS HALF OF THE STORY

TOBIAS KOPPERS

TOBIAS KOPPERS

- GITHUB: [@sokra](#)
- TWITTER: [@wSokra](#)
- 2012: I CREATED WEBPACK.
- SINCE 2017: I WORK FULL-TIME ON WEBPACK AS PART OF THE CORE TEAM.

Platinum Sponsors

Platinum Sponsors are those who have pledged \$50,000 or more to webpack.



Gold Sponsors

Gold Sponsors are those who have pledged \$10,000 to \$50,000 to webpack.



WEBPACK 5

- **ME:** YEAH. WEBPACK 5 IS COMING
- **MAYBE YOU:**
 - YEAH. WEBPACK 5 IS COMING.
 - OH NO THERE ARE BREAKING CHANGES.
 - THIS MAY BREAK MY BUILD.
 - I NEED TO INVEST TIME TO FIX IT.
 - WHY ARE THERE BREAKING CHANGES ANYWAY...?

THE “WHY”

- THIS TALK IS ABOUT THE **WHY** OF THE LARGEST BREAKING CHANGES
- **BONUS:** YOU GET TO **KNOW** THE BREAKING CHANGES. JUST TO BE **PREPARED**.
- **BONUS:** YOU SEE THE **BENEFITS** OF THESE CHANGES

MAJOR RELEASING

- PEOPLE TEND TO DISLIKE BREAKING CHANGES
- NOBODY IS FORCED TO UPGRADE.
 - BUT I THINK IT'S WORTH IT!

Tobias Koppers
@wSokra

How should be the major release cycle of webpack?

Option	Percentage
1 month or less	6%
2-3 months	24%
4-6 months	47%
Show answers	22%

709 votes · Final results

9:55 PM · Jul 20, 2017 · Twitter Web Client

NON-BREAKING RELEASES

WEBPACK 4

- 4.0.0, 4.0.1, 4.1.0, 4.1.1, 4.2.0, 4.3.0, 4.4.0, 4.4.1, 4.5.0, 4.6.0, 4.7.0, 4.8.0, 4.8.1, 4.8.2, 4.8.3, 4.9.0, 4.9.1, 4.9.2, 4.10.0, 4.10.1, 4.10.2, 4.11.0, 4.11.1, 4.12.0, 4.12.1, 4.12.2, 4.13.0, 4.14.0, 4.15.0, 4.15.1, 4.16.0, 4.16.1, 4.16.2, 4.16.3, 4.16.4, 4.16.5, 4.17.0, 4.17.1, 4.17.2, 4.17.3, 4.18.0, 4.18.1, 4.19.0, 4.19.1, 4.20.0, 4.20.1, 4.20.2, 4.21.0, 4.22.0, 4.23.0, 4.23.1, 4.24.0, 4.25.0, 4.25.1, 4.26.0, 4.26.1, 4.27.0, 4.27.1, 4.28.0, 4.28.1, 4.28.2, 4.28.3, 4.28.4, 4.29.0, 4.29.1, 4.29.2, 4.29.3, 4.29.4, 4.29.5, 4.29.6, 4.30.0, 4.31.0, 4.32.0, 4.32.1, 4.32.2, 4.33.0
- 33 MINOR RELEASES
- 43 PATCH RELEASES
- BUT: TECHNICAL DEPT PILLS UP...

STRATEGY TO PREVENT MAJOR RELEASES

- HACKS, AND ADD **TODOS** TO THE SOURCE CODE
- OPT-IN FLAGS
- **PLAN AHEAD FOR FUTURE FEATURES**
 - MAKE NECESSARY **PREPARATIONS** IN MAJOR RELEASES
- RESTRICT API TO BE ABLE TO DO CHANGES IN **NON-BREAKING WAY**
 - VERY DIFFICULT, BECAUSE OF THE **LARGE PLUGIN API SURFACE**

PART I: MOTIVATION

LONG TERM GOALS

- TO ANSWER THE “WHY” WE NEED TO HAVE LONG TERM GOALS
- VOTING PAGE → FEATURES
- TECHNICAL DEPT → CLEANUP WORK
- PREPARATION → REFACTORING, MODERNIZATION

 webpack

DOCUMENTATION CONTRIBUTE VOTE BLOG     

Vote and Prioritize

Tobias Koppers // sokra 
824 ● 519 ○

This mini-application allows you to browse and vote on new features for webpack. Log in with your GitHub credentials and you will notice that you have a certain amount of points/influence that can be used to vote for or against any of the features listed below. The following two sections describe the different types of influence and how they can be attained.

Influence ⚙

Influence is a unit of measure based on time you have been a member on GitHub. However, from 2017 on you will receive one influence per day.

Golden Influence ⚙

Golden Influence is equal to 100 *normal influence*. Golden Influence is obtained by being a backer or sponsor on our [Open Collective page](#).

[WEBPACK 5] PERSISTENT CACHE	PLACE YOUR VOTE
Evaluate how webpack could cache intermediate results on disk. This will reduce the initial build time when restarting webpack. Status: Partially handled by the cache-loader.	 22114 x 1 +  2914 x 100 313514
Sponsors	PLACE YOUR VOTE
Coming soon...	

[WEBPACK 5] CONSISTENT HASHING	PLACE YOUR VOTE

webpack.js.org/vote

VOTING PAGE: PERSISTENT CACHE

- INCREMENTAL BUILDS IN WATCH MODE ARE FAST
- BUT: RESTARTING WEBPACK CAN BE SLOW
- PERSISTENT CACHE: PROMISES BETTER BUILD STARTUP PERFORMANCE
- CACHE ON DISK (OR OTHER PERSISTENT MEDIUMS)

VOTING PAGE: CONSISTENT HASHING

- PEOPLE HAVE DIFFICULTIES USING HASHING FOR LONG TERM CACHING
- WEBPACK CHANGES FILE CONTENT MORE OFTEN THAN EXPECTED
- CONSISTENT HASHING: PROMISES EASIER AND MORE RELIABLE HASHING.

PART II: BREAKING CHANGES

FOCUS ON NODE.JS MODULES?

- WEBPACK USED TO HAVE THE GOAL TO MAKE NODE.JS MODULES (NPM) WORK IN THE BROWSER
- THIS FOCUS SHIFTED. MORE AND MORE FRONTEND MODULES WHERE PUBLISHED.
- YOU CAN STILL USE `require("events")` AND `import "crypto"` IN WEBPACK. EVEN `node_modules` CAN USE THAT.

NODE CORE MODULE POLYFILLS

- PROBLEM: THESE NODE.JS CORE MODULE POLYFILLS ARE OFTEN **LARGE**
- MINIMIZED SIZE:
 - CRYPTO: 301KB
 - ZLIB: 128KB
 - STREAM: 63KB
 - READABLE-STREAM: 62KB
 - STRING_DECODER: 27KB
 - BUFFER: 23KB

NODE CORE MODULE POLYFILLS

- PROBLEM: DEVELOPERS OFTEN UNAWARE OF THE USED NODE CORE MODULE POLYFILLS.
- PROBLEM: IT'S TOO EASY TO ACCIDENTALLY INCLUDE THEM.
- PROBLEM: POLYFILLS DIFFER FROM NODE CORE MODULES DUE TO BROWSER LIMITATIONS.
- PROBLEM: THE NATIVE WEB PLATFORM DOESN'T SUPPORT THEM.
- PROBLEM: THESE MODULES RESERVE NAMES (I. E. constants)
- BREAKING CHANGES IN NODE CORE MODULES RESULT IN BREAKING CHANGES IN POLYFILLS AND NEED TO RESULT IN A MAJOR WEBPACK RELEASE
- PROBLEM: POLYFILLS ARE OFTEN OUTDATED.

NODE CORE MODULE POLYFILLS REMOVED

- MIGRATION:
 - AVOID USING NODE CORE MODULES FOR FRONTEND CODE
 - PREFER FRONTEND-COMPATIBLE MODULES FROM NPM
 - OPT-IN TO USE POLYFILLS BY INSTALLING THEM MANUALLY AND ALIAS IN CONFIG
- EFFECTS:
 - USERS WILL TRY TO AVOID THEM
 - USERS ARE AWARE OF USED NODE POLYFILLS
 - USERS CONTROL VERSION OF POLYFILL USED
 - INSTALL SIZE OF WEBPACK DECREASES

FRONTEND LANDSCAPE CHANGES

- WEB APPLICATIONS ARE MUCH LARGER COMPARED TO YEARS AGO.
- FRONTEND TEAMS ARE LARGER.
- FRONTEND IS DEPLOYED MORE OFTEN.



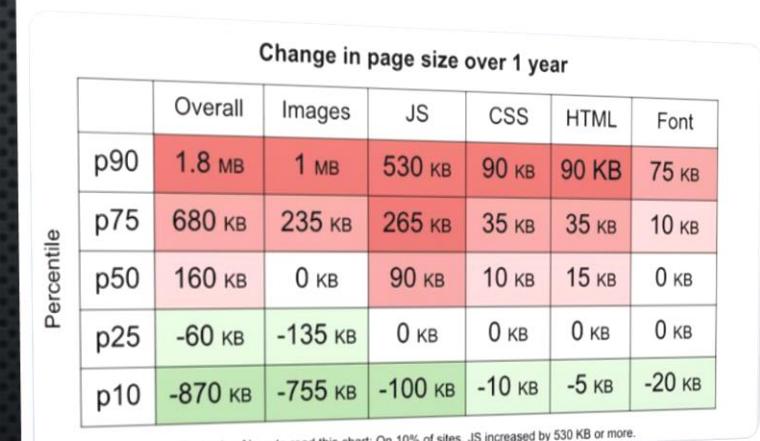
Katie Hempenius
@katiehempenius

How many sites* are getting larger?

Over 1 year:

70% of sites got larger.
47% of sites increased JS by 100 KB+.
18% of sites grew by 1 MB+.

* Alexa Top 10K sites, July 2018 vs July 2017, via [@HTTPArchive](#)



5:11 AM · May 28, 2019 · Twitter Web Client

http://t.co/58V6NlMA1f2

b10	-810 KB	-102 KB	-100 KB	10 KB	-2 KB	-50 KB
b25	-60 KB	-130 KB	0 KB	0 KB	0 KB	0 KB

FOCUS ON BUNDLE SIZE?

- WEBPACK OPTIMIZES FOR BUNDLE SIZE BY DEFAULT.
- PROBLEM: OPTIMIZATION BY SIZE IS A WHOLE APPLICATION OPTIMIZATION AND POTENTIALLY INVALIDATES ALL ASSETS OF THE WEBAPP.
- AVOIDING THIS IS DIFFICULT AND REQUIRED A LOT OF KNOWLEDGE AND CONFIGURATION.
- WEBPACK 4 WAS NOT OPTIMIZED FOR THAT.

NEW FOCUS: LONG TERM CACHING

- NEW DETERMINISTIC ALGORITHMS TO OPTIMIZE FOR LONG TERM VALIDITY OF OUTPUT FILES
- OVERHEAD:
 - UP TO 800 MODULES: $\min(4n, 1k - n) \rightarrow \text{MAX: } 800b$ AT 200 MODULES
 - UP TO 8000 MODULES: $\min(5n, 10k - n) \rightarrow \text{MAX: } 8kb$ AT 1.6K MODULES
 - UP TO 80,000 MODULES: $\min(6n, 100k - n) \rightarrow \text{MAX: } 80k$ AT 14K MODULES
- NOTE: NUMBER OF EMITTED MODULES < NUMBER OF INPUT MODULES (SCOPE HOISTING)

DETERMINISTIC BY DEFAULT

- MIGRATION:
 - CHECK IF LONG TERM CACHING IS RELEVANT FOR YOU. (HINT: IT PROBABLY IS.)
 - YOU MIGHT NEED TO OPT-IN FOR SIZE OPTIMIZATION INSTEAD IF YOU CARE MORE FOR BUNDLE SIZE THAN LONG TERM CACHING.
 - CALCULATE THE DIFFERENCE.

PERSISTENT CACHING

- WEBPACK ALREADY HAS AN IN-MEMORY CACHING SYSTEM FOR INCREMENTAL BUILDS
- HOW TO PERSIST ITS CONTENT?
- WHAT'S IN THE CACHE? MAINLY **Module** OBJECTS.

SERIALIZATION

- PROBLEM: Module OBJECTS ARE NOT SERIALIZABLE.
 - THERE ARE SUBCLASSES
 - THERE ARE NON-PRIMITIVE NESTED OBJECTS (I. E. Dependencies, Sources)
 - ETC.
- IMPLEMENT A **SERIALIZATION SYSTEM** WHICH CAN HANDLE THIS.
- MAKE **Module** AND OTHER CLASSES **SERIALIZABLE**.

MODULE GRAPH

- PROBLEM: THE MODULE GRAPH IS PART OF THE `Module` CLASS.
- SPLIT `ModuleGraph` INTO SEPARATE CLASS TO MAKE `Module` REUSABLE BETWEEN COMPILATIONS

ASYNC CACHING

- PROBLEM: CACHING INTERFACE IS **SYNCHRONOUS** AND VERY LIMITED,
BECAUSE IS WAS CREATED FOR SINGLE USE CASE: IN-MEMORY CACHING
- CHANGE INTERFACE TO BE **ASYNC** AND EXTENSIVE VIA PLUGINS

COST OF CACHING

- PROBLEM: PERSISTENT CACHING IS MORE EXPENSIVE DUE TO SERIALIZATION AND DISK
→ RUN PERSISTENT CACHING WHEN COMPILER IS IDLE OR CLOSING
- PROBLEM: THE PUBLIC COMPILER API HAS NO NOTATION ABOUT CLOSING/DISPOSING
→ ADD A `close` METHOD THAT SHOULD BE CALLED

MODERNIZATION

- INCREASED MINIMUM NODE.JS VERSION TO 8
- NEW DATA-STRUCTURES
 - ARRAYS → Sets
 - OBJECTS → Maps
- CHOKIDAR (WITH NATIVE fsevents DEP) → OWN PURE-JS WATCHING

PREPARATION

- NESTED CALLBACKS + SEMAPHORES → QUEUES
- STATS FACTORY → STATS PLUGINS + DEFAULT PLUGINS
- BOOLEAN FOR USED/PROVIDED EXPORTS → CLASS WITH DETAILED INFO
- COMMON RUNTIME → RUNTIME MODULES INJECTED WHEN USED

THANKS

SLIDES:

<https://github.com/sokra/slides>

Feature

BONUS: OTHER NEW FEATURES

NEW FILENAMES FOR DEVELOPMENT

- PROBLEM: WEBPACK GENERATES USELESS FILENAMES (1.bundle.js, 2.bundle.js)
- NEW ALGORITHM FOR USEFUL FILENAMES
- `import("./lazy-loaded")` → `src_lazy-loaded_js.bundle.js`
- `splitChunks` → `react_react-dom.bundle.js`

NEW FILENAMES BY DEFAULT

- MIGRATION:
 - MAKE SURE TO BE ABLE TO WORK WITH THE DIFFERENT FILENAMES.
 - IN MOST CASES THERE IS NOTHING TO DO.
 - YOU CAN OMIT `/* webpackChunkName: "name" */` NOW.

EXTENSIBLE STATS

- THE STATS ARE NOW EXTENSIBLE BY PLUGINS
- LOOK FORWARD TO NEW KINDS OF PLUGINS ADDING ADDITIONAL INFORMATION/OUTPUT

EXTENSIBLE HOT MODULE REPLACEMENT

- HOT MODULE REPLACEMENT IS NOW EXTENSIBLE BY PLUGINS
- THIS ALLOWS TO ADD HMR FOR OTHER MODULE TYPES THAN JS
 - I. E. CSS, WASM, LOCALIZATION, HTML

NESTED TREE-SHAKING

- TREE-SHAKING IS NOW ABLE TO TRACK REEXPORTED NAMESPACE OBJECTS
- `import { b } from "a"; b.x();`
- A.JS: `import * as b from "b"; export { b };`
- B.JS: `export { x, y, z };`

THANKS

SLIDES:

<https://github.com/sokra/slides>

BONUS: PREPARED FEATURES

EXPERIMENTS

- WE HAVE NOW CONFIG OPTIONS TO TOGGLE EXPERIMENTS
 - EXPERIMENTS WILL HAVE RELAXED SEMVER, MAY BREAK IN MINOR VERSIONS
 - THIS WILL ALLOW US TO EXPERIMENT IN MINOR VERSIONS
 - AND MOVE FASTER WITH NEW STUFF
 - WHILE MOVING SLOWER WITH MAJOR VERSIONS
 - WATCH CHANGELOGS CLOSELY WHEN USING EXPERIMENTS

PARALLELIZATION

- WE ARE WORKING TOWARDS USING MULTIPLE WORKERS FOR A SINGLE COMPILED PROGRAM
- ON THE SAME COMPUTER VIA WORKER_THREADS OR CHILD PROCESSES
- ON MULTIPLE COMPUTERS VIA NETWORK

MODULE TARGET

- WE ARE WORKING TOWARDS COMPILING A BUNDLE WITHOUT TOP-LEVEL RUNTIME
- CONSUMABLE AS ESM

NEW SPECS

- WE ARE EXPERIMENTING WITH NEW WEB SPECS
 - CSS MODULES (THE SPEC ONE, NOT THE EXISTING ONE)
 - HTML MODULES
 - WASM-ESM-INTEGRATION
 - TOP-LEVEL-AWAIT

THANKS

SLIDES:

<https://github.com/sokra/slides>