

MY STORY WITH WEBPACK

TOBIAS KOPPERS

GITHUB: `@sokra`

TWITTER: `@wSokra`

THE STORY

FOLLOWED BY “NOW” AND “THE FUTURE”

2008

- STUDIED COMPUTER SCIENCE
- JAVA
- DISCOVERED GOOGLE WEB TOOLKIT (GWT)
 - JAVA TO JAVASCRIPT COMPILER
 - GENERATES BIG JS FILES
 - BUT ALSO HAD A SOLUTION FOR THIS



GWT CODE SPLITTING

```
Button b = new Button("Click me", new ClickHandler() {
    public void onClick(ClickEvent event) {
        GWT.runAsync(new RunAsyncCallback() {
            public void onFailure(Throwable caught) {
                // ...
            }
            public void onSuccess() {
                Window.alert("Hello world");
            }
        });
    }
});
```

2011

- BACHELOR THESIS
- MY FIRST CONTACT WITH OPEN SOURCE
- PRESENTATION WITH WEB TECHNOLOGY
- JMPRESS.JS: A jQUERY PLUGIN FOR PRESENTATIONS ON A INFINITE CANVAS BY `@shama`
 - CONTRIBUTED A LOT, GOT A MAINTAINER
 - JUST CHECKED IT FOR THIS PRESENTATION: IT'S BROKEN NOW...
`transition: --transform?`
 - THERE ARE BETTER SOLUTIONS NOW ANYWAY

2012

- MASTER THESIS
- PART OF IT WAS A WEB-APPLICATION
 - NOTE: I LIKE PREMATURE OPTIMIZATION
 - I WAS LOOKING FOR A “BUNDLER”
- I FOUND modules-webmake BY @medikoo
 - BUT IT WAS LACKING CODE SPLITTING
 - SO I OPENED AN ISSUE...

2012: ISSUE IN MODULES-WEBMAKE

- INITIALLY I PROPOSED TO “ABUSE” `process.nextTick` FOR CODE SPLITTING
 - THAT WAS DUMB...
 - BUT THE IDEA WAS BORN
- SOME PEOPLE THOUGHT THIS IS AN USELESS FEATURE
 - “HOW BIG IS YOUR WEBAPP GOING TO GET?”
 - WELL ACTUALLY WEBAPPS ARE GOING TO GET **VERY BIG** THESE DAYS

2012: PROTOTYPE

- AFTER SOME DISCUSSION I IMPLEMENTED A PROTOTYPE OF MY IDEA, FROM SCRATCH
- IT USES `require.ensure` FOR CODE SPLITTING
- IT ALSO USES NUMERIC IDS, INSTEAD OF RESOLVING THE REQUEST AT RUNTIME
 - SMALLER BUNDLE
 - BUT COMES WITH LIMITATIONS...

2012: WEBPACK

- MY INITIAL IDEA WAS TO ADD THIS TO `modules-webmake`
- BUT BECAUSE IT WAS A COMPLETELY NEW APPROACH:
 - I WAS ASKED TO RENAME THE PROJECT
 - `sokra/modules-webpack`
 - LATER `webpack/webpack`

2012: DYNAMIC REQUIRES

- DUE TO THE NUMERIC IDS `require(someExpression)` WAS NO LONGER POSSIBLE
- COMMONJS MODULES WERE PRETTY DYNAMIC THESE DAYS
- ADDED THE CONTEXT FEATURE
 - DYNAMIC EXPRESSIONS CAN BE PROCESSED LIKE `require("./folder/" + expression)` OR `require(expression)`
 - ALSO AVAILABLE AS `require.context(...)`
 - RESOLVES ALL POSSIBLE REQUESTS AT BUILD-TIME AND EMBEDS MAPPINGS INTO A CONTEXT MODULE

2012: CONTINUED DEVELOPMENT

- ADDED POLYFILLS FOR SOME NODE.JS MODULES
 - MORE WILL BE ADDED LATER
- ADDED LOADERS
 - style-loader
 - css-loader
 - ...



2012: FIRST CONTRIBUTOR

- **@shama** FIXED SOME TYPOS IN THE DOCS (PR #1)

2012: CONTINUED DEVELOPMENT

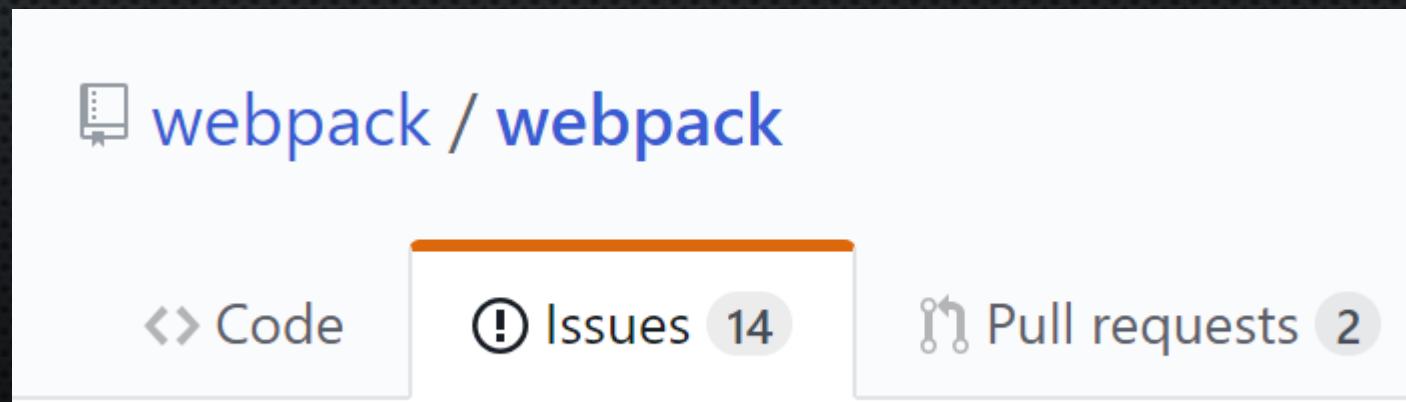
- [hash] SUPPORT
- file-loader
- --watch
- @shezi FIXED A BUG (PR #2)
 - SEEMS LIKE PEOPLE ACTUALLY USING IT

2012: CONTINUED DEVELOPMENT

- IN-MEMORY CACHING
- MORE COMMONJS FEATURES
 - `require.resolve`
 - `require.cache`
- `debug` OPTION
 - EQUALS TO `devtool: "eval"` NOW

JUL 2012: ISSUES?

- @jhnnns OPENED ISSUES #3 - #16



2012: CONTINUED DEVELOPMENT

- ALIGN WITH NODE.JS COMMONJS
- ADDED AMD SUPPORT
 - IMPORTANT FEATURE FOR REQUIRE.JS MIGRATION
- `webpack-dev-middleware/server`

2012: MULTI-THREADING

- MULTI-THREADING
 - YES, THIS WAS ACTUALLY A FEATURE IN 0.x
 - IT WAS LATER REMOVED
 - OVERHEAD OF INTER-PROCESS-COMMUNICATION WAS TOO BIG
 - HAD NO PERFORMANCE BENEFIT IN GENERAL
 - COMPLEXITY

2012: ENHANCED-REQUIRE

- THERE ALSO WAS A MODULE NAMED “enhanced-require”
 - EMULATED WEBPACK FEATURES FOR NODE.JS
 - `require.ensure`
 - LOADERS
 - IT WAS LATER ABANDONED IN FAVOR OF COMPILING BEFORE RUNNING
 - `target: "node"`

JAN 2013: BIG REFACTORING FOR 0.9

- ADDED A PLUGIN SYSTEM CALLED `tapable`
- REFACTORED ALL FEATURES INTO PLUGINS
- CREATED STILL EXISTING ARCHITECTURE:
 - Compilation
 - Compiler
 - ...

2013: CONTINUED DEVELOPMENT

- `require.include`
- PR #3 BY `@jhnnns` WHICH FIXES SOMETHING
 - MORE PRs FOLLOWING

JUN 2013: HOT MODULE REPLACEMENT

- I ASKED FOR IDEAS
- GOT SOME FEEDBACK
- *@Phoscur*
“HOLY GRAIL: HOT CODE RELOAD. REPLACE MODULE OBJECTS AT RUNTIME.”

2013: NPM AND NODE_MODULES

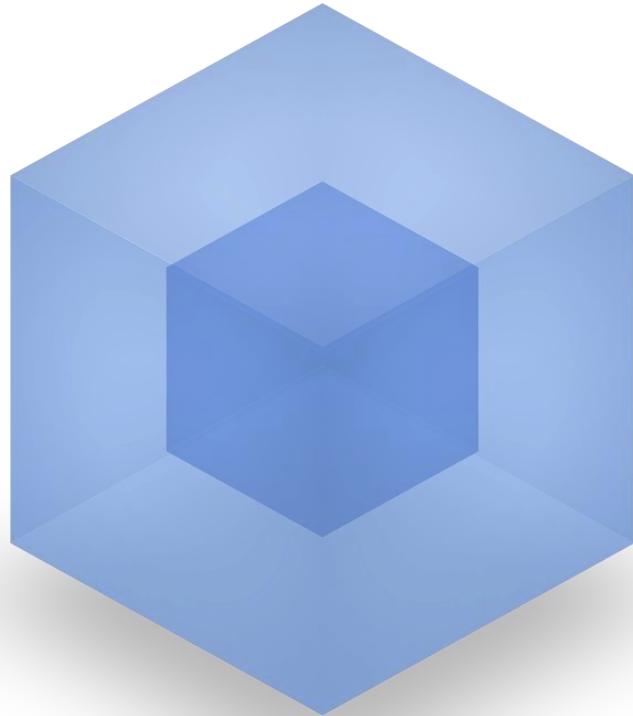
- WEBPACK HAD A `DedupePlugin` THESE DAYS
 - IT FINDS AND REUSES `DUPLICATE MODULE CODE`
 - NPM CREATED A `DEEP node_modules` STRUCTURE THESE DAYS
 - THAT WAS A BIG PROBLEM FOR FRONTEND DEVELOPMENT
 - THE PLUGINS WAS LATER DROPPED IN FAVOR OF `npm dedupe`

2013: BROKEN MODULES

- WEBPACK HAS A LOT OF FEATURES TO “FIX” MODULES
 - `var x = require; x("module");`
 - `!function(r) { r("module"); }`
`(typeof require === "function" ? require : function() {...})`
 - MODULES WERE REALLY DIFFICULT TO STATICALLY ANALYZE THESE DAYS
 - IT GOT BETTER...
 - AND EVENTUALLY FIXED WITH ECMASCRIPT MODULES

VERSION 1

- DEC 2013: FIRST BETA VERSION OF WEBPACK 1
- FEB 2014: 1.0.0 RELEASED



webpack
MODULE BUNDLER

JUL 2014: REACT-HOT-LOADER

- DAN ABRAMOV ASKED ABOUT HOT MODULE REPLACEMENT ON STACKOVERFLOW
 - AND CREATED react-hot-loader
 - HE WAS LATER HIRED BY FACEBOOK, TO WORK ON REACT.JS

JUL 2014: INSTAGRAM.COM

- PETE HUNT: OSCON 2014: How INSTAGRAM.COM WORKS
 - Instagram.com USES WEBPACK BECAUSE OF IT'S CODE SPLITTING FEATURE
 - AND THEY ALSO LIKED THE SHARED DEPENDENCY GRAPH FOR JS, CSS, IMAGES, ETC.
 - I DIDN'T RECOGNIZED THIS AS FEATURE BEFORE
 - THEY USES A FORKED VERSION OF `require.js` BEFORE
 - NOW WEBPACK WAS FAMOUS!

JS

MAR 2015: HARMONY

- FIRST PROTOTYPE OF HARMONY MODULES SUPPORT
 - HARMONY MODULES = Now EcmaSCRIPT MODULES (ESM)
 - webpack-2 BRANCH

2015

- ADDED SUPPORT FOR NON-JS CONFIGS
 - PRs BY `@monder` AND `@akre54`, USING `interpret`
 - YOU CAN NOW WRITE `webpack.config.coffee`
 - `coffee-script` WAS HIP
 - NOW BETTER:
 - `webpack.config.ts`
 - `webpack.config.babel.js`

NOV 2015: FIRST WEBPACK 2 BETA

- CONFIGURATION IS NOW VALIDATED BY **SCHEMA**
 - OH THIS AVOIDS **A LOT** OF ISSUES ABOUT **WRONG** CONFIGURATIONS
 - WEBPACK IGNORED ANY INCORRECT OPTION BEFORE THAT...
- ADDED PERFORMANCE HINTS
- NEW **DOCUMENTATION**
 - HANDLED BY ITS OWN TEAM
 - I'M PRETTY BAD AT WRITING DOCUMENTATION (PROVE → SEE WEBPACK 1 DOCS)

NOV 2015: FIRST WEBPACK 2 BETA

- ECMASCRIPT MODULES ARE NOW NATIVELY SUPPORTED
- TREE SHAKING (UNUSED EXPORTS ELIMINATION)
- INTERNAL CODE IS IN PROGRESS OF BEING UPGRADED TO ES2015

2016: ORGANIZATION

- @TheLarkInn JOINED AND HELPED A LOT WITH PUBLIC RELATIONSHIP
- JUN 2016: CORE TEAM CREATED
- OCT 2016: JOINED OPENCOLLECTIVE TO GET DEVELOPMENT FUNDED
- OCT 2016: JOINED JS FOUNDATION TO STAY OPEN AND FREE

2017

- JAN 2017: SPLIT INTO TWO ORGANIZATIONS
 - WEBPACK AND WEBPACK-CONTRIB
 - EASIER TO MANAGE
 - SEPARATE TEAMS
- JAN 2017: WEBPACK 2 RELEASED
 - IT WAS ACTUALLY WEBPACK 2.2 FOR SEMVER REASONS/MISSTAKES...
 - 2 YEARS AND 11 MONTHS AFTER VERSION 1



webpack

- WE ALSO GOT A NEW LOGO

2017: FULL-TIME & FIRST BIG SPONSORS

- MAR 2017: AG-GRID STARTED SPONSORING (FOR A BANNER IN THE DOCUMENTATION)
- APR 2017: I WENT FULL-TIME WEBPACK
 - WE SAVED A BIT OF MONEY IN OPENCOLLECTIVE
 - I'VE GOT A 3-MONTH SPONSORSHIP BY [SEGMENT](#)
 - IT WAS A BIT RISKY, NOT ENOUGH MONTHLY MONEY FOR FULL TIME AT THIS TIME

JUN 2017: WEBPACK 3

- WEBPACK 3 RELEASED
 - SCOPE HOISTING (MODULE CONCATENATION)
 - NO BIG BREAKING CHANGES
 - MOSTLY INTERNAL CHANGES
 - 5 MONTHS AFTER VERSION 2



AUG 2017:

оbeaLiJd

- TRIVAGO STARTED SPONSORING WITH \$10k/MONTH
 - PRETTY IMPORTANT FOR WEBPACK
- THIS SECURED MY FULL-TIME WORK ON WEBPACK
 - SO PRETTY IMPORTANT FOR ME PERSONALLY TOO

JAN 2018: ALPHA VERSION WEBPACK 4

- PERFORMANCE IMPROVEMENTS
 - IMPROVED ALGORITHMS
 - IMPROVED PLUGIN SYSTEM
- USABILITY IMPROVEMENTS
 - `mode` FOR PRODUCTION/DEVELOPMENTS DEFAULTS
 - `splitChunks` INSTEAD OF `CommonsChunkPlugin`

JAN 2018: ALPHA VERSION WEBPACK 4

- OPTIMIZATION IMPROVEMENTS
 - BETTER TREE SHAKING WITH THE `sideEffects` FLAG
- MODULE TYPES: NATIVE SUPPORT FOR:
 - JS (CJS/AMD/ESM)
 - JSON
 - WASM (WEBASSEMBLY MODULE)

JAN 2018: ALPHA VERSION WEBPACK 4

- `webpack-cli`
 - NOW A SEPARATE MODULE
 - SEPARATE TEAM
 - CAN INNOVATE SEPARATELY. CAN MAJOR RELEASE SEPARATELY.
 - REDUCED INSTALL SIZE WHEN NOT USING THE CLI:
 - `create-react-app`, `angular-cli`, `vue-cli`, `preact-cli`, ...

FEB 2018: WEBPACK 4 LEGATO

- 4.0.0
 - WE FOLLOWED THE RELEASE CYCLE TOO STRICTLY (6 MONTHS)
 - ECOSYSTEM WAS NOT READY
 - DOCUMENTATION WAS NOT READY
 - A LOT OF COMPLAINS... LESSONS LEARNED FOR v5
 - APART FROM THIS A PRETTY SUCCESSFUL VERSION



NOW

FOLLOWED BY "THE FUTURE"

NOW: STILL WEBPACK 4?

- YES, BUT WE ADDED A LOT OF NEW FEATURES WITHOUT BREAKING CHANGE:
 - `splitChunks.maxSize`
 - `/* webpackPrefetch: true */`
 - WEBASSEMBLY IMPROVEMENTS
 - `DefinePlugin.runtimeValue`
 - ...
- ALSO A LOT OF BUGFIXES, TESTS, LINTING, CI, ETC.

SPLITCHUNKS.MAXSIZE

- MAIN.js
- VENDORS~MAIN.js

```
optimization: {  
  splitChunks: {  
    minSize: 1000,  
    maxSize: 10000  
  }  
}
```

- MAIN~B2C7414A.js
- MAIN~6E7EAD72.js
- MAIN~052B3814.js
- VENDORS~MAIN~3FF27526.js
- VENDORS~MAIN~11485824.js
- VENDORS~MAIN~75F09DE8.js
- VENDORS~MAIN~6A2AE26B.js
- VENDORS~MAIN~17ACAD98.js
- VENDORS~MAIN~7274E1DE.js

PREFETCH

- `import(/* webpackPrefetch: true */ "module")`
 - DOWNLOADS CHUNK
 - WHEN IMPORT IS EXECUTED
 - OR WHEN IDLE AND IMPORT IS VISIBLE
 - EVALUATES CHUNK
 - WHEN IMPORT IS EXECUTED

DEFINEPLUGIN.RUNTIMEVALUE

```
new DefinePlugin({
  COMMIT: DefinePlugin.runtimeValue(
    () => {
      fs.readFileSync(".git/HEAD", ...) // ...
      return commit;
    },
    [".git/HEAD", ".git/info/refs"].map(p => path.resolve(p))
  )
})
```

THE FUTURE

WEBPACK 5?

- WORK IN PROGRESS
 - “next” BRANCH ON WEBPACK/WEBPACK
- TRY IT WITH
 - "webpack": "webpack/webpack#next" IN package.json
 - YARN RESOLUTIONS FOR NESTED DEPENDENCIES
 - GIVE FEEDBACK



Branch: next ▾

New pull request

This branch is 306 commits ahead, 27 commits behind master.

WEBPACK 5 GOALS

- REFACTORING... A LOT... ✅ (done, for now)
- MINIMIZE BREAKING CHANGES, BUT DEPRECATE OLD THINGS ✭ (partial)
 - I DON'T WANT TO WRITE A LONG MIGRATION GUIDE
- REMOVE ALREADY DEPRECATED THINGS ✅ (done)
- DON'T CARE ABOUT RELEASE CYCLE ✭ (partial)
 - RELEASE WHEN DONE + MIGRATION GUIDE + SOME IMPORTANT PLUGINS WORK
 - NEED TO DO A `iswebpack5ready` CHECKLIST...

WEBPACK 5 GOALS

- IMPROVE PERFORMANCE WITH PERSISTENT CACHING
 - #1 ON VOTING PAGE
 - ADD ABILITY TO (DE)SERIALIZE MODULES ✅ (done)
 - ADD A `FileCachePlugin`, NEXT TO THE `MemoryCachePlugin` ✅ (done)
 - LOAD AND STORE MODULES TO THE FILESYSTEM ✅ (done)
 - INVALIDATION IS THE CHALLENGE... ❌ (not started)

WEBPACK 5 GOALS

- IMPROVE USABILITY OF LONG TERM CACHING
 - RELATED TO #2 ON VOTING PAGE
 - CHANGE DEFAULTS TO **HTTP/2** ✘ (not started)
 - PROVIDE GOOD ALTERNATIVES TO “records”
 - **records** ARE TOO DIFFICULT TO USE
 - DETERMINISTIC ALGORITHMS FOR IDS ✘ (partial)

WEBPACK 5/6 MORE IDEAS

- NAMED RULES

```
module.rules: {  
    css: { test: /\.css$/, use: ... }  
}
```

- EASIER TO OVERRIDE
- MAY REMOVE `module.defaultRules`
- ARRAY IS STILL SUPPORTED
- SAME FOR PLUGINS

WEBPACK 5/6 MORE IDEAS

- MULTI-THREADING
 - BASED ON THE (DE)SERIALIZATION NEEDED FOR PERSISTENT CACHING
- REFACTOR RUNTIME GENERATION
 - ADD `RuntimeModules`
 - FIXES PROBLEMS WITH HMR WHEN RUNTIME CHANGES
 - NEEDED STEP FOR NO-RUNTIME OUTPUT
- WASM IMPROVEMENTS FOR THE NEW SPEC

WEBPACK 5/6 MORE IDEAS

- REFACTOR STATS TO PLUGINS
 - ALSO ALLOW EXTENDING STATS WITH PLUGINS
- REFACTOR HMR FOR ALLOW HMR FOR OTHER MODULE TYPES, I. E. CSS
- `import.meta`
- ALIGN WITH NODE.JS `.mjs` WORK

WEBPACK 5 RELEASE DATE

- DON'T ASK!

THE END

... OF THIS PRESENTATION, BUT NOT OF THE WEBPACK STORY

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