

WEBPACK

Chunking and lazy loading
@ajaegle #jscc17

ABOUT ME

- Andreas Jäggle
- Software Engineer at **AXOOM**
- Web, Java, Spring, TypeScript, Angular, Build-Tooling
- Stream Processing, Infrastructure, Docker
- @ajaegle

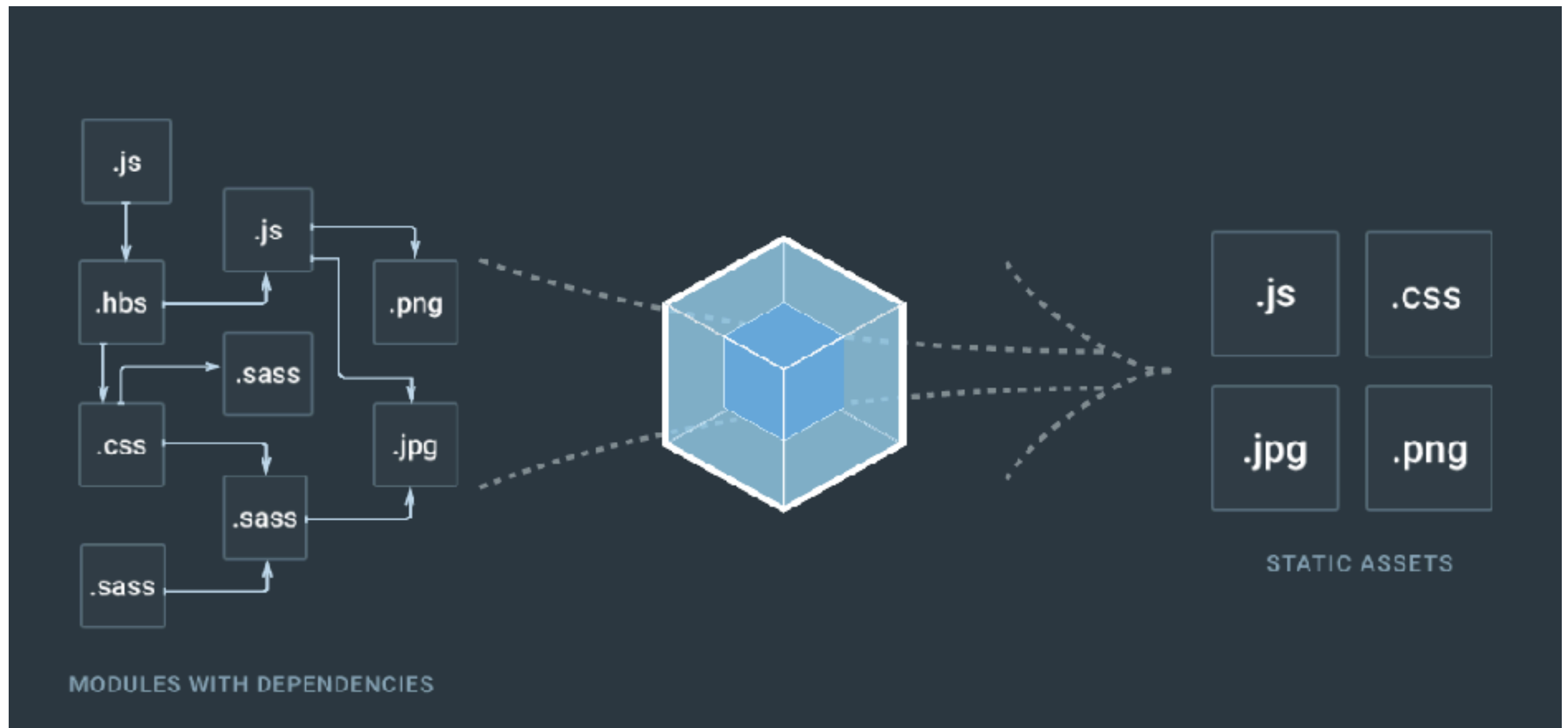
WEBPACK

- Module loader / bundler
- Every asset is transformed to a module:
JavaScript, CSS, Images, Templates, JSON, ...
- Traversing a tree of dependencies instead of concatenating a bunch of JavaScript files
- Itself heavily modularised for extensibility



DEPENDENCY TREE

.....



Source: <https://webpack.js.org/>

ENTRIES, LOADERS, PLUGINS

- Entrypoints
 - Starting points of tree traversal
- Loaders
 - Regex Test
 - Transformation on matched file
 - Chainable (e.g. scss -> css -> style)
- Plugins
 - Global functions including more than one unit
 - e.g. uglify, minify

CACHING AND CHUNKING

CACHING

- Problem: How long can bundle.js be cached?
- Use checksum when generating bundle file names
bundle-[hash].js
- Cache your assets forever!
- Let webpack add the script tags
 - HtmlWebpackPlugin

BUNDLE SIZE – COMMONS CHUNK

- Bundles can get pretty large (vendor libs, assets, ...)
- Extract common stuff into an own bundle
 - CommonsChunkPlugin
 - name-[chunkhash].js
- WARN: don't forget manifest (the webpack runtime)

DEFER IMPORTS

- Not everything is required on first load
- Webpack supports `System.import()` syntax
- `System.import("./mymodule").then(
 module => module.thefunction()
)`
- Internally used in Angular Lazy Modules
 - `Router.forChild()` and `loadChildren`

DEMO

FURTHER READING

- The webpack docs, really!
- <https://webpack.js.org/>
- Some online courses:
- <http://www.webpack.academy/>
- <https://www.udemy.com/webpack-2-the-complete-developers-guide/>