WEBPACK

Minification

Sreekanth M. E.

Freelance Trainer & Consultant

http://www.SreekanthME.com

Minification is the process of removing all unnecessary characters from source code without changing its functionality.

The main purpose is to reduce the size of source file(s) without changing behavior.

Smaller files load faster in the browser thus providing performance benefits.

Some Minification actions:

- Removing white space characters
- Removing new line characters
- Removing comments
- Truncating variable & function names to single characters

For humans, minified code is unreadable.

But for computers, both minified and un-minified code are equal.

Minification cannot be reversed as variable and function names are changed during minification.

However, white spaces and new lines can be added to make minified code pretty (or readable).

Minification can be achieved in the following ways:

- Running webpack -p
- Using UglifyJsPlugin

WEBPACK -P

webpack -p is equivalent to webpack --optimize-minimize --define process.env.NODE_ENV = ‘production’

It performs the following steps:

- Minification using UglifyJsPlugin
- Sets the NodeJS env variable to 'production' triggering certain packages to compile differently.

UGLIFYJSPLUGIN

```
const webpack = require('webpack')


module.exports = {
  devtool: 'source-map',
  plugins: [
    new webpack.optimize.UglifyJsPlugin({sourceMap: true})
  ]
}
```

SOURCE MAP

A source map provides a way of mapping code within a minified file back to it’s original position in the source file.

Makes reading & debugging minified files in browser easy.

Both CSS and JS files can have a source map.

Webpack config object has a key named devtool which controls if and how source maps are generated.

There are multiple source mapping types available in webpack but the highest quality one is named source-map.

Webpack source map types:

- Inline source maps that reside inside bundle files.
- Separate source maps that reside in separate files.

Source map files contain, at the beginning, a JSON object with information about the map itself and the original source files.

Minification programs add a special comment pointing to the source map file at the end of the minified files.

Example:

```
//# sourceMappingURL=/path/to/script.js.map
```

Source maps will only work in the browser if:

- Support for source maps is enabled in the browser.
- Web server supports source maps.

Source map files are downloaded ONLY when developer tools are open in the browser.

Source Maps are enabled by default in chrome.

Settings that control behavior associated with source maps are under settings → sources tab of chrome dev tools:

- Enable JavaScript source maps
- Enable CSS source maps

If these settings are changed, reload page and reopen dev tools.

If source map is enabled in chrome and if source maps are available, the console error messages point to line numbers in the original source files.

Else console error messages point to line numbers inside the bundle file(s).

Source Maps cause dev tools to load original files in addition to minified ones.

Originals files can be used to set breakpoints and step through code. Meanwhile, browser is actually running minified code behind the scenes.

Source maps give the illusion of debugging source files.

END OF CHAPTER

APPENDIX