

| | |
|--|-----------|
| How to use this book | 4 |
| Part 1 - The webpack tutorial | 5 |
| The basics | 5 |
| Why webpack? | 5 |
| Code along: create a complete webpack app | 7 |
| Your first webpack app | 7 |
| Babel and webpack.config.js | 10 |
| Your first webpack.config.js | 11 |
| Loaders to transform anything you want | 12 |
| Use babel loader to transform ES6 | 13 |
| Babel React | 16 |
| First of all: create the React code | 16 |
| Configure Babel so that it can transpile JSX | 17 |
| Make webpack understand the .jsx extension | 17 |
| Features for your app | 18 |
| Styling | 18 |
| Part 2 - Production-ready apps with webpack | 21 |
| What happens under-the-hood | 21 |
| Webpack CLI | 21 |
| Everything starts with the entry file | 22 |
| Loader | 24 |
| Compiler seals the dependency graph | 25 |
| Manifest | 26 |
| Plugins | 26 |
| Output file | 28 |
| NPM and package.json | 28 |
| Installing NPM. | 28 |
| Creating a new project with NPM | 28 |
| The package.json file | 29 |
| The fields dependencies and devDependencies | 30 |
| Global dependency | 31 |
| Scripts | 31 |
| package-lock.json | 33 |

| | |
|--|----|
| NPX | 33 |
| The folder structure of a webpack project | 34 |
| The HTML file. | 34 |
| Resolver extensions | 35 |
| Creating production-ready apps with webpack | 36 |
| Separate prod and dev configs | 37 |
| Remove duplicated code with webpack-merge | 39 |
| More than one output | 44 |
| How to make two bundles - public and admin | 44 |
| How to create one server bundle and one client bundle | 46 |
| Another option: both configs in the same file | 49 |
| Create an awesome dev experience | 50 |
| webpack-dev-server | 51 |
| Get started with webpack-dev-server | 52 |
| Hot reloading with React | 54 |
| Hot reloading under-the-hood | 56 |
| Optimized production builds to get a small bundle size | 58 |
| Run webpack in production mode | 58 |
| Code splitting | 59 |
| Caching | 59 |
| Code splitting for caching | 61 |
| How to implement code split for caching | 63 |
| Code split vendors | 63 |
| Code split CSS | 65 |
| Code splitting for dynamic imports | 66 |
| How to implement dynamic code splitting | 67 |
| What kind of code splitting should you use? | 71 |
| Tree shaking | 72 |
| Use import statements - not require keyword | 72 |
| Import by module name | 73 |
| Babel config | 73 |
| Prefer dependencies that are ES6 modules | 74 |
| Special tweaks for some dependencies | 74 |
| Are you using lodash? | 74 |
| Are you using moment.js? | 75 |
| Bundle analyzer | 76 |
| Better debugging with source maps | 76 |

| | |
|---|-----------|
| What problem does source maps solve? | 76 |
| Source maps under-the-hood | 78 |
| More fine-grained control for source maps | 79 |
| Enable source maps in production | 80 |
| More options for dev source maps | 81 |
| Part 3 - Recipes | 83 |
| Base webpack setup | 83 |
| React | 84 |
| Vue | 86 |
| Typescript | 88 |
| Styles | 91 |
| CSS | 91 |
| CSS modules | 92 |
| Combining CSS modules and CSS | 93 |
| PostCSS | 94 |
| SASS | 95 |
| LESS | 96 |
| Stylus | 97 |
| Images | 97 |
| SVG | 98 |
| PNG | 98 |