

Table of Contents

[Introduction 1](#_Toc159779110)

[References 3](#_Toc159779111)

[Course Contents 3](#_Toc159779112)

[GitHub List Commits 3](#_Toc159779113)

[What Even Is Webpack 3](#_Toc159779114)

[Installing and Running Webpack and Webpack-CLI 5](#_Toc159779115)

[Install Webpack 7](#_Toc159779116)

[Webpack Default Configuration 9](#_Toc159779117)

# Introduction

**Webpack** is a powerful tool used in web development. Let’s dive into what it does:

1. **Module Bundler**:
   * **Webpack** is primarily a **module bundler**.
   * Its main purpose is to bundle JavaScript files (and other assets) for usage in a web browser.
   * Imagine it as a way to package all your code and assets into a single file or a few files that the browser can efficiently load.
   * It takes your project’s various modules (JavaScript files, CSS, images, etc.) and combines them into a single output file (or multiple files) that can be served to the browser.
2. **JavaScript Transformation and Optimization**:
   * Besides bundling, **Webpack** can also perform transformations on your code.
   * It can transpile modern JavaScript (ES6+) into older versions (ES5) using tools like Babel.
   * It optimizes your code by minifying it (removing unnecessary whitespace and renaming variables) to reduce file size.
   * This results in faster loading times for your web application.
3. **Configuration-Driven**:
   * **Webpack** uses a configuration file called webpack.config.js.
   * In this file, you define how your project should be bundled.
   * You specify entry points (where the bundling process starts), output paths, loaders (for handling different file types), and plugins (for additional tasks like code splitting or generating HTML files).
4. **Loaders and Plugins**:
   * **Loaders** allow **Webpack** to process different types of files (e.g., CSS, images, fonts) and convert them into valid modules.
   * **Plugins** enhance the bundling process by performing additional tasks.
   * For example, the **HTMLWebpackPlugin** generates an HTML file with the bundled JavaScript automatically injected.
5. **Code Splitting**:
   * **Webpack** supports code splitting, which means breaking your bundle into smaller chunks.
   * This is useful for optimizing performance by loading only the necessary parts of your application when needed (e.g., lazy loading routes).
6. **Development and Production Modes**:
   * **Webpack** has different modes: **development** and **production**.
   * In development mode, it provides features like source maps for easier debugging.
   * In production mode, it aggressively optimizes your code for better performance.
7. **Community and Documentation**:
   * The **Webpack** community is active, and there are plenty of plugins and loaders available.
   * The official documentation is comprehensive and provides guides, concepts, and examples to get you started.

In summary, **Webpack** is a versatile tool that simplifies managing your project’s assets, optimizes your code, and makes web development more efficient. If you’re new to it, check out the [Getting Started guide](https://webpack.js.org/guides/getting-started/) to explore its features!

# References

[Learn Webpack - Full Tutorial for Beginners](https://www.youtube.com/watch?v=MpGLUVbqoYQ&list=WL&index=3&t=2341s)

[Colt Steele's YouTube channel](https://www.youtube.com/channel/UCrqAGUPPMOdo0jfQ6grikZw)

[Colt Steele Github Code](https://github.com/Colt/webpack-demo-app)

Course Contents

⌨️ (0:00:00) What Even Is Webpack??

⌨️ (0:08:12) Installing and Running Webpack and Webpack-CLI

⌨️ (0:22:18) Imports, Exports, & Webpack Modules

⌨️ (0:29:58) Configuring Webpack

⌨️ (0:38:57) Loaders, CSS, & SASS

⌨️ (0:53:55) Cache Busting and Plugins

⌨️ (1:07:13) Splitting Dev & Production

⌨️ (1:17:13) Html-loader, File-loader, & Clean-webpack

⌨️ (1:28:17) Multiple Entrypoints & Vendor.js

⌨️ (1:34:45) Extract CSS & Minify HTML/CSS/JS

# GitHub List Commits

1. [Initial app code, no webpack](https://github.com/johncusey/webpack-demo-app/commit/b96a9dbce0d4abbf156612ccb23e4be74d6fd313) (What Even Is Webpack??)
2. [Break code into separate scripts, no webpack](https://github.com/johncusey/webpack-demo-app/commit/dd5492493336955c66c8960d4e50d76297199fbb) (Installing and Running Webpack and Webpack-CLI)

# What Even Is Webpack

Webpack Official Website:

A screenshot of a computer

Description automatically generated

Webpack Official GitHub Website:

A screenshot of a computer

Description automatically generated

The file structure webpack generates.

A screenshot of a computer

Description automatically generated

Command for building a React Application.

A close up of a sign

Description automatically generated

The file structure Webpack generates

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

# Installing and Running Webpack and Webpack-CLI

Example code base used by Colt Steele

A screenshot of a computer

Description automatically generated

The file structure of the starter project.

A screenshot of a computer

Description automatically generated

Screen shot the the appliction

A screenshot of a web page

Description automatically generated

A screenshot of a web page

Description automatically generated

The coding structure after refactoring the code into many files for Webpack to bundle.

A screenshot of a computer

Description automatically generated

You have to include all the JavaScript files in script tags. If you use Webpack you can eliminate all these script tags.

A screen shot of a computer program

Description automatically generated

## Install Webpack

1) Install Package Json

Install package json file run the following command. The package.json file is a crucial part of Node.js projects. Here’s what it does:

1. **Metadata and Descriptive Information**:
   * It’s a **JSON file** residing in the root directory of your project.
   * Contains both **human-readable metadata** (like project name and description) and **functional metadata** (such as package version number).
   * Helps people discover your package by providing a **description** and **keywords**.
   * [Specifies the **project homepage URL** and the **issue tracker URL** where users can report issues1](https://docs.npmjs.com/cli/v6/configuring-npm/package-json/)[2](https://heynode.com/tutorial/what-packagejson/).
2. **Dependency Management**:
   * Lists the **dependencies** required by your application.
   * When you run npm install, it installs the dependencies specified in this file.
   * Helps manage versions and ensures consistent behavior across different environments.
3. **Versioning**:
   * The **name** and **version** fields together form a unique identifier for your package.
   * Changes to the package should correspond to changes in the version.
   * If you plan to publish your package, these fields are **required**.
   * [If not publishing, they are **optional**1](https://docs.npmjs.com/cli/v6/configuring-npm/package-json/).
4. **License Information**:
   * Specify a **license** for your package (e.g., BSD-3-Clause or MIT).
   * Helps users understand how they can use your package.
   * [Use SPDX license identifiers or custom strings1](https://docs.npmjs.com/cli/v6/configuring-npm/package-json/).
5. **Scripts and Commands**:
   * Define custom **scripts** (e.g., build, test, start) that can be executed using npm run.
   * Makes it easy to automate common tasks during development.
6. **Other Details**:
   * You can include a **file** named <filename> at the top level of the package.
   * [Deprecated styles like license objects are now replaced with SPDX expressions1](https://docs.npmjs.com/cli/v6/configuring-npm/package-json/)[3](https://nodesource.com/blog/the-basics-of-package-json/).

In summary, package.json is your project’s **manifest**, containing essential information for Node.js, npm, and JavaScript development.

A screenshot of a computer error

Description automatically generated

**$ npm init -y**

2) Install Webpack and Webpack CLI



**npm install --save-dev webpack webpack-cli**

3) Changing script in Package Json

A screen shot of a computer program

Description automatically generated

# Webpack Default Configuration

Have Webpack bundle the project by using the script in the Package Json. The **“scripts”** property allows you to define custom commands that can be executed.



If you see the following error you need ***index.js*** file in the ***src folder***.

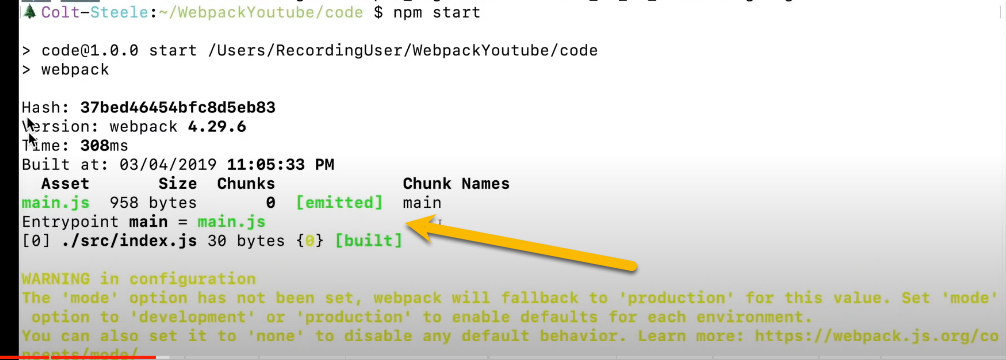


Solution:

A screen shot of a computer

Description automatically generated

After run Webpack you get the following message. The ***index.js*** is the entry point

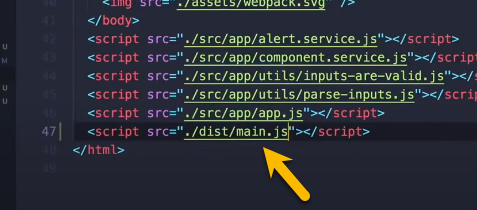


Webpack create the **dist** folder with ***main.js***

A screen shot of a computer code

Description automatically generated

You need to added ***main.js*** to your ***index.html*** file.



Now you can see calling the ***index.js*** entry point file.

A screenshot of a computer

Description automatically generated