

Webpack 5 Documentation

Personal Details

- Name: Rachit Anand Srivastava
- Institute: Thapar Institute of Engineering and Technology
- Email: rac.sri25@gmail.com
- Contact: +91 7071690418
- Country of Residence: India
- Time Zone : IST (GMT + 05:30)

Technical Knowledge

I am a sophomore at Thapar Institute of Technology pursuing Computer Engineering as my major field. I work in the field of Web/App development and Blockchain. My specialization is in Full Stack development. I have done two internships, one in my freshman year in a Company named *Started India*, and the other (currently ongoing) in my second year in a company named *Zuzu.AI* as an extension developer. I am working in the field of MERN Stack for almost 1.5 years now and am able to create production-ready products on my own.

Resources that I have referred to learn and grow includes various courses from Udemy, Coursera, Frontend masters. I also read a lot of articles online and I always try to keep up with the latest technology via platforms like LinkedIn.

Some of my projects include:

- Farmer Portal: An application built to connect farmers and dealers directly and eliminate any middleman in between to maximize the farmer's earning. Goods price (which are generally decided by auction in India) was decided using an

auction system using Blockchain. The entire application was made using Node, ReactJs, MongoDB.

- Funding India: An application built to manage the payments done to all the different political parties. The purpose was to eliminate the collection of black money by the different parties. Stack used was: Node, ReactJs, MongoDB, Ethereum.
- A chatbot using NodeJS for Facebook messenger.
- A movie service system using ReactJs and NodeJS.
- E-Health care: System designed to ease the communication between doctors and patients for faster delivery of healthcare services.
- A video calling platform to perform court trials in case the presence of the witness etc. needed in the courtroom is difficult.
- An app to keep track of your daily exercises: using React Native.
- A Bookstore system using NodeJS.
- A Blog System implemented using NodeJS.
- A ping pong game made using vanilla JavaScript.
- Currently making a smart suggestion Extension using ReactJs.
- Currently making my own Instagram Clone using React Native and Redux.

Languages/Frameworks/Runtime Environment/Libraries/Tools I am proficient at :

- JavaScript
- Java
- NodeJS
- ReactJs
- React Native
- Webpack
- GraphQL
- Ethereum
- Sockets
- Redux (+ contextApi)

Reach Me:

- rac.sri25@gmail.com
- github.com/rachit2501
- linkedin.com/in/rachit-anand-srivastava-345307173/

About Project

The browsers currently don't support modules. There are several bundlers that came into existence before webpack. Some of the popular alternatives to webpack are Browserify, Parcel etc. Webpack is a module bundler which enables the developer to write and maintain his code using modules, and then ship them as a single bundle so that the end client can view the final product. In current market, webpack is the most powerful and preferred module bundler.

SO WHY WEBPACK?

Webpack has surpassed all other tools that were introduced to achieve similar outcome. If we consider some of the most popular tools like gulp, grunt, browserify, all of them were left in dirt the moment power of webpack was realized across the globe. Some major points of webpack which makes it the bundler of choice are:

- **HOT-MODULE RELOAD:** The webpack dev server comes with a hot module reloading feature making developing process a bliss.
- Advance control over bundles using concepts like **code splitting**, splitting of bundles into smaller pieces resulting in faster loading on the client side by shipping minimal code.

- The concepts of **PLUGINS**: Plugins is, in my opinion one of the most powerful feature of webpack. Plugins is stated in the official documentation as a JavaScript object with an apply function. Plugins operate over our entire project as a whole resulting in more power in the hands of the developer.
- **ELIMINATION OF DEAD CODE**: Webpack can find code that you are not actually using and not add it to the final bundle. This results in removal of unnecessary code consequently reducing the bundle size. Hence when shipped, the loading experience becomes much optimized.
- Code written in ES6 and above can easily be transpiled to ES5 using Babel.
- **LOADERS**: Another feature of webpack. Allows to preprocess different types of files, allowing us to bundle any static resource, way beyond JavaScript, in our node project and shipping them with our bundle.
- **PRESETS/ADDONS**: According to documentations “*webpack-presets* provides an abstraction over top of regular Webpack configuration. Even though this adds some complexity, it also allows you to implement shareable presets.” This shows the additional power the developer gets while analyzing his code for shipping.

Webpack is shipped pre-configured with frameworks like angular and development tools like “Create-React-App”. I have worked on webpack in a few projects. My project *Lottery System* has one of the few setups of webpack that I have built from scratch. Besides, I have also studied the create-react-app webpack configuration and modified it to make the configuration allowing development of extensions over react. I have over 5 months of experience with working with webpack. I am familiar with how webpack-4 works exactly, even have idea of like how the bundling process works in case of webpack.

Timeline

- **PreGSoC** : I’d focus on getting involved with the community. Get detailed information about the details about the features, differences between the

webpack5 and its ancestors. Study the exact working of webpack and analyze how the new webpack release improves on the formally executed algorithms.

- **Week 1:** Start creating the first draft. Create a basic structure of what all is needed in the documentation and in what order. Write the introduction content of the documentation and table of content.
- **Week 2-4:** This week I will work on writing the basic installation guide. After that I will focus on writing about the different concept like plugins, loaders etc. involved with webpack.
- **Week 5:** Details about the power webpack offers to the developer like code splitting, format support etc.
- **Week 6:** Analyze the current draft and improve its structuring.
- **Week 7-8 :** Make a separate draft about the new features in Webpack 5.
- **Week 9-10 :** Make another separate draft about the main key differences between webpack5 and its former versions.
- **Week 11:** In the first draft, finish up the draft by writing about the ways to contribute and details on how to support the project.
- **Week 12-13:** Compile and interlink all the 3 drafts and finalize my first prototype. Then spend the remaining time left on improvising the documentation. I will be seeking community guidance more actively during this period.

SOME FINAL THOUGHTS:

I am confident that I will be able to well document this project owing to the fact that I am someone with the firsthand experience about how webpack works. I am someone who has started his webpack journey under the guidance of one of the core members of webpack [Sean T. Larkin](#) , thanks to Frontend Masters.