

# Precache Files with Webpack

This page explains how to use the `workbox-webpack-plugin` node module to generate the list of files to precache and add it to your service worker.

## Install `workbox-webpack-plugin`

Start by installing `workbox-webpack-plugin` from npm.

```
$ npm install workbox-webpack-plugin --save-dev
```



## Setup your Webpack Config

To add the manifest to your service worker file you'll need to add the plugin to your `webpack.config.js` file like so:

```
// Inside of webpack.config.js:
const WorkboxPlugin = require('workbox-webpack-plugin');

module.exports = {
  // Other webpack config...

  plugins: [
    // Other plugins...

    new WorkboxPlugin.InjectManifest({
      swSrc: './src/sw.js',
    })
  ]
};
```



If you look at the output of your service worker file you'll see an extra file is added as an `importScripts()` call. It'll be named `precache-manifest.<revision>.js`. This file will contain the list of files to precache and it will expose the list as the variable `self.__precacheManifest`.

## Precache Files

The final step is to tell Workbox to precache the files, which you can do by adding the following code to your service worker.

```
workbox.precaching.precacheAndRoute(self.__precacheManifest || []);
```



This will precache any of the files from the Webpack plugin.

## Further Reading

The `workbox.precaching` API provides some options for configuration should you need to customize any of the default behaviors. [Learn more on the workbox.precaching page.](#)

---

*Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 3.0 License](#), and code samples are licensed under the [Apache 2.0 License](#). For details, see our [Site Policies](#). Java is a registered trademark of Oracle and/or its affiliates.*

*Last updated April 3, 2018.*