# **0x00. Webpack**

**Front-endJavaScriptES6**

* Weight: 1
* Project will start Oct 28, 2024 6:00 AM, must end by Oct 29, 2024 6:00 AM
* Checker was released at Oct 28, 2024 12:00 PM
* An auto review will be launched at the deadline



## **Resources**

**Read or watch**:

* [Webpack documentation](https://intranet.alxswe.com/rltoken/XEFTUAcZ_9sKurp1Bui7ug)
* [Webpack beginner guide](https://intranet.alxswe.com/rltoken/6ngQzrV7xeKJjcRwdmrYAQ)
* [npm-package.json](https://intranet.alxswe.com/rltoken/P00rJM5qCeaf33hsPuhgog)

## **Learning Objectives**

At the end of this project, you are expected to be able to [explain to anyone](https://intranet.alxswe.com/rltoken/ME_HaE8ycyyO7RaAoOycIw), **without the help of Google**:

* How to setup Webpack for a basic project
* Entry points, output, and loaders
* How to add plugins
* How to split your code into chunks
* How to setup a dev server

## **Requirements**

* All of your code will be executed on Ubuntu 18.04 LTS using Node 12.x.x
* Allowed editors: vi, vim, emacs, Visual Studio Code
* All of your files should end with a new line
* A README.md file at the root of the folder of the project is mandatory

## **Tasks**

### **0. Basic setup**

**mandatory**

Create and run Webpack using a basic installation.

* Create a folder named task\_0.
* Install webpack and webpack-cli as developer dependencies within the folder using npm.
* Install jQuery as a regular dependency using npm.
* Make sure that webpack and webpack-cli are listed under the devDependencies key along with jQuery being listed under the dependencies key within the package.json
* Create a src directory with a index.js in it.
* The file should import jquery and add three different paragraphs to the page body: (refer to this [link](https://intranet.alxswe.com/rltoken/rOv-DBm6Vx5OxBKyKp_69Q) under the Babel section for importing jQuery)

Holberton Dashboard

Dashboard data for the students

Copyright - Holberton School

* Create a dist/index.html. Import your main.js in the body.
* You must use jQuery to add the elements to the body of the page.
* When running Webpack, your javascript and html files should be generated in a dist folder.
* You should not use a custom webpack config file.
* Opening your main file should not generate any error in the console.
* Your html code should only import one Javascript script (the one generated by webpack).
* Do NOT push your dist/main.js if you have one.

**Repo:**

* GitHub repository: alx-react
* Directory: 0x00-Webpack
* File: task\_0/package.json, task\_0/src/index.js, task\_0/dist/index.html

Check submission

### **1. Learning how to use Webpack with a config file**

**mandatory**

**Installing packages**

* Create a folder named task\_1, cd into it, and create a package.json using npm init -y
* Install webpack (dev dependency), jQuery (dependency), and Lodash (dependency) within the folder using npm.
* Modify your package.json to add a build script that runs webpack to create a production build. This lets you execute npm run build on the command line.

**Using jQuery**

* Create a js directory with a javascript file named dashboard\_main.js in it. The file should import jquery and add the below elements in the following order:
  + Paragraph element: Holberton Dashboard
  + Paragraph element: Dashboard data for the students
  + Button element with the text Click here to get started
  + Add a paragraph element as so <p id='count'></p>
  + Add another paragraph: Copyright - Holberton School

**Prevent spammers**

* Write a function called updateCounter() that tracks the number of times the button element has been clicked.
* Each time it’s called, update the count by 1 and set the content of the paragraph element with id=‘count’ to {count} clicks on the button
* Bind the debounce function in Lodash to the click event on the button you created previously.

**Requirements:**

* When running Webpack, your javascript should be generated in a public folder. You should also place your index.html in this folder.
* The javascript file generated by Webpack should be named bundle.js
* Set Webpack config mode to production
* Opening your html file should not generate any error in the console
* Your html code should only import one Javascript script (the one generated by webpack)

**Resources:**

* [Debounce documentation on Lodash](https://intranet.alxswe.com/rltoken/fuAiFYn08OQLb7SM8cYdYA)

**Repo:**

* GitHub repository: alx-react
* Directory: 0x00-Webpack
* File: task\_1/js/dashboard\_main.js, task\_1/package.json, task\_1/webpack.config.js, task\_1/public/index.html

Check submission

### **2. Adding CSS & Images**

**mandatory**

Create a specific configuration for Webpack.

* Using the folder named task\_2.
* Reuse the code from task\_1.
* Modify the webpack config to support adding CSS to the bundle.
* Modify the webpack config to support adding images to the CSS.
* Create a folder named css
* In a file named main.css, change the position of the counter text to be displayed on the right of the button and in bold.
* Add an element at the top of document with the id #logo which has a width of 200px and height of 200px.
* Set the background of th element with the image in task\_2/assets/holberton-logo.jpg (you can download this [link](https://intranet-projects-files.s3.amazonaws.com/holbertonschool-webstack/581/holberton-logo.jpg)).
* Set the width and height of the logo background image to 200px by 200px.
* Make sure to configure Webpack to optimize images.
* When running Webpack, your javascript and html files should be generated in a public folder.
* The javascript file generated by Webpack should be named bundle.js.
* Set Webpack config mode to production.
* Opening your main file should not generate any error in the console.
* Your html code should only import one Javascript script (the one generated by webpack).
* When running Webpack, you should not see the warning WARNING in asset size limit: The following asset(s) exceed the recommended size limit

**Repo:**

* GitHub repository: alx-react
* Directory: 0x00-Webpack
* File: task\_2/package.json, task\_2/css/main.css, task\_2/webpack.config.js, task\_2/js/dashboard\_main.js, task\_2/public/index.html

Check submission

### **3. Dev servers, modules, and tree shaking**

**mandatory**

Using the folder named task\_3. **Set up a development server:**

* Reuse the code from task\_2.
* Modify the Webpack config to setup a development server running on the port 8564.
* Modify the Webpack config to set its mode to development.
* Add a script in the package.json to start the server and open the browser with npm run start-dev.

**Divide the code into modules:**

* Divide the main bundle into three modules.
* header should contain a header.css and header.js files.
* Import jQuery, and add the logo and the H1 title to the header.js file (with content Holberton Dashboard). Add a console.log printing Init header.
* Add the needed style to the header.css file.
* body should contain a body.css and body.js files.
* Import jQuery, Lodash and add the body code (button, counter) in the body.js file.
* Add the needed style to the body.css file.
* footer should contain a footer.css and footer.js files.
* Import jQuery, and append a paragraph the copyright text Copyright - Holberton School.
* Add the needed style to the footer.css file.
* Modify the Webpack configuration to support three different entry points (header, body, footer). Each entry point should generate a filename with the following format name\_of\_the\_file.bundle.js.
* Do NOT have a task\_3/public/ directory pushed to your repository.
* Add a plugin to Webpack to automatically create an index.html HTML file

**Improve development speed**

* Modify the webpack config to support inline source mapping.
* Check that the console.log in the header.js now takes you to your javascript file instead of the main bundle.
* Add a plugin to Webpack to clean your build folder on each build.

**Improve bundles size:**

* You can see that the current modules generated with npm run build are quite large. They all contain jQuery and/or Lodash. Modify the Webpack configuration to split the modules in chunks.

**Requirements:**

* When running the dev server, your code should work on http://localhost:8564/.
* When running Webpack, your javascript and html files should be generated in a public folder.
* Opening your main file should not generate any error in the console.

**Repo:**

* GitHub repository: alx-react
* Directory: 0x00-Webpack
* File: task\_3/modules/body/body.css, task\_3/modules/body/body.js, task\_3/modules/footer/footer.css, task\_3/modules/footer/footer.js, task\_3/modules/header/header.css, task\_3/modules/header/header.js, task\_3/package.json, task\_3/webpack.config.js

Check submission

Copyright © 2024 ALX, All rights reserved.