

Lesson 05 Demo 06

Working with Webpack and Modern JavaScript

Objective: To demonstrate the process of configuring and using Webpack to bundle and manage modern JavaScript applications efficiently

Tools required: Visual Studio Code and Node JS

Prerequisites: None

Steps to be followed:

1. Write a JavaScript program for Webpack
2. Execute and verify the working of Webpack

Step 1: Write a JavaScript program for Webpack

- 1.1 Execute the following command to check the node version:

```
node -version
```



A screenshot of a terminal window. The window has tabs at the top: PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The TERMINAL tab is selected. The terminal shows the following text:
● labuser@ip-172-31-41-18:~/Desktop\$ node --version
v12.22.9
○ labuser@ip-172-31-41-18:~/Desktop\$ █

- 1.2 Create a folder **webpack**, write the following command to create the folder, and navigate to the Webpack directory:

```
mkdir webpack
```

```
cd webpack
```



A screenshot of a terminal window. The terminal shows the following text:
● labuser@ip-172-31-41-18:~/Desktop\$ mkdir webpack
● labuser@ip-172-31-41-18:~/Desktop\$ █

```
● labuser@ip-172-31-41-18:~/Desktop$ cd webpack
○ labuser@ip-172-31-41-18:~/Desktop/webpack$ █
```

1.3 Initialize **package.json** by running the following command:

npm init

```
○ labuser@ip-172-31-41-18:~/Desktop/webpack$ npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help init` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (webpack)
version: (1.0.0)
description:
entry point: (index.js)
test command:
git repository:
keywords:
author:
license: (ISC)
About to write to /home/labuser/Desktop/webpack/package.json:

{
  "name": "webpack",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {}
```

1.4 Write the following command to install the Webpack Command Line Interface (CLI):

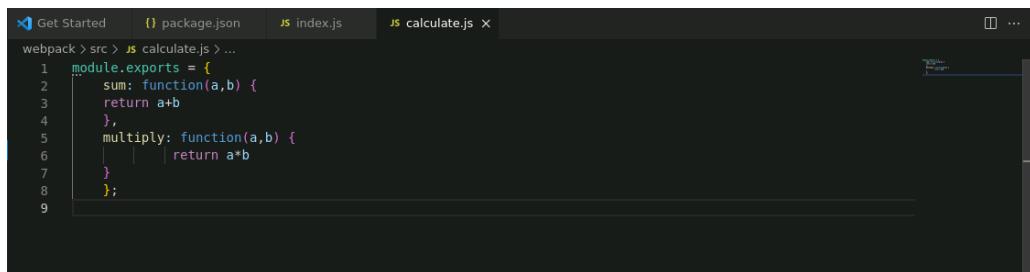
npm install webpack webpack-cli --save-dev

```
Is this OK? (yes)
● labuser@ip-172-31-41-18:~/Desktop/webpack$ npm install webpack webpack-cli --save-dev
npm WARN EBADENGINE Unsupported engine {
  npm WARN EBADENGINE   package: 'webpack-cli@0.0.1',
  npm WARN EBADENGINE   required: { node: '>=18.12.0' },
  npm WARN EBADENGINE   current: { node: 'v12.22.9', npm: '8.5.1' }
}
npm WARN EBADENGINE Unsupported engine {
  npm WARN EBADENGINE   package: '@discoveryjs/json-ext@0.6.3',
  npm WARN EBADENGINE   required: { node: '>=14.17.0' },
  npm WARN EBADENGINE   current: { node: 'v12.22.9', npm: '8.5.1' }
}
npm WARN EBADENGINE Unsupported engine {
  npm WARN EBADENGINE   package: '@webpack-cli/configtest@3.0.1',
  npm WARN EBADENGINE   required: { node: '>=18.12.0' },
  npm WARN EBADENGINE   current: { node: 'v12.22.9', npm: '8.5.1' }
}
npm WARN EBADENGINE Unsupported engine {
  npm WARN EBADENGINE   package: '@webpack-cli/info@3.0.1',
  npm WARN EBADENGINE   required: { node: '>=18.12.0' },
  npm WARN EBADENGINE   current: { node: 'v12.22.9', npm: '8.5.1' }
}
npm WARN EBADENGINE Unsupported engine {
  npm WARN EBADENGINE   package: '@webpack-cli/serve@3.0.1',
  npm WARN EBADENGINE   required: { node: '>=18.12.0' },
  npm WARN EBADENGINE   current: { node: 'v12.22.9', npm: '8.5.1' }
```

Step 2: Execute and verify the working of Webpack

2.1 Create a folder named **src** and a JavaScript file inside it named **calculate.js** add the following code to it:

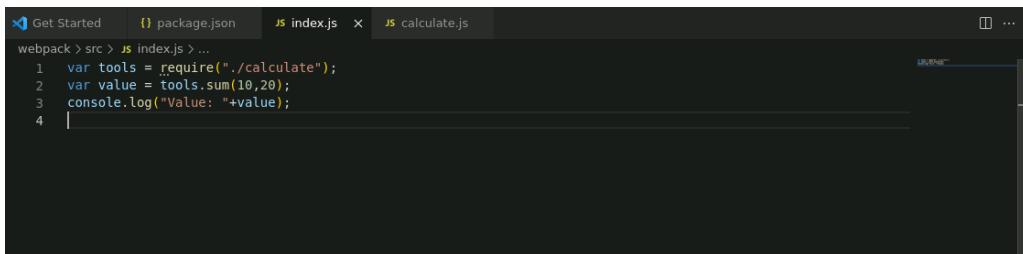
```
module.exports = {
  sum: function(a,b) {
    return a+b
  },
  multiply: function(a,b) {
    return a*b
  }
};
```



A screenshot of a code editor showing the contents of the calculate.js file. The file contains the code provided in the previous text block. The code defines a module with two exports: 'sum' and 'multiply'. The 'sum' function adds two numbers, and the 'multiply' function multiplies them. The code is written in ES6 syntax, including arrow functions.

2.2 Create another JavaScript file named **index.js** and add the following code to it:

```
var tools = require("./calculate");
var value = tools.sum(10,20);
console.log("Value: "+value);
```



A screenshot of a code editor showing the contents of the index.js file. The file contains the code provided in the previous text block. It imports the tools module from calculate.js and uses its sum function to calculate the value of 30, then logs "Value: 30" to the console.

2.3 Open **package.json** and add the following build script:

```
"scripts": {  
  "build": "node src/index.js"  
}
```



```
webpack > {} package.json > {} scripts > build  
1  {  
2    "name": "webpack",  
3    "version": "1.0.0",  
4    "description": "",  
5    "main": "index.js",  
6    "Debug":  
7    "scripts": {  
8      "test": "echo \\\"Error: no test specified\\\" && exit 1",  
9      "build": " node src/index.js"  
10    },  
11    "author": "",  
12    "license": "ISC",  
13    "devDependencies": {  
14      "webpack": "^5.75.0",  
15      "webpack-cli": "^5.0.1"
```

2.4 Run the build using the following command:

```
npm run build
```

```
○ labuser@ip-172-31-41-18:~/Desktop/webpack$ npm run build
```

```
> webpack@1.0.0 build  
> node src/index.js  
Value: 30
```

By following the above steps, you have successfully configured and used Webpack to bundle and manage modern JavaScript applications efficiently, ensuring modular code organization and streamlined execution.