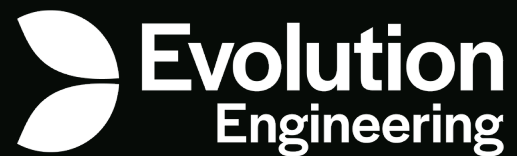


Webpack



Что такое вебпак?



Ключевые концепции

- Entry
- Output
- Loaders
- Plugins
- Mode

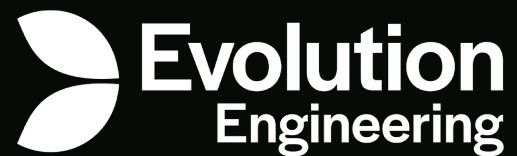
Entry

Точка входа откуда webpack начнет строить граф зависимостей
модулей проекта:
`/src/index.js`

Output

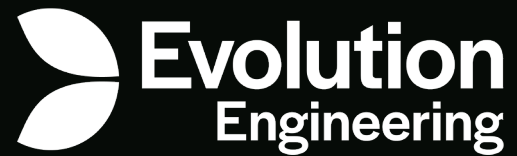
Куда собрать проект:

- `/dist/main.js`
- `./dist/`



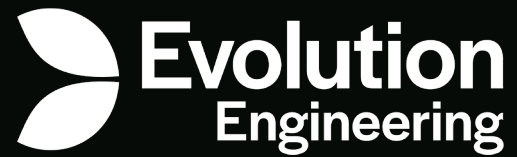
Loaders

Преобразуют всё кроме javascript/json в валидные webpack модули
– CSS, картинки, svg



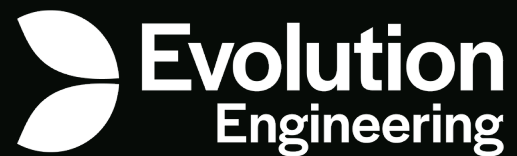
Plugins

- Optimization
- Minification
- Установка переменных окружения



Mode

- Development
- Production
- None



Поехали

```
mkdir webpack-demo  
cd webpack-demo  
npm init -y
```


Устанавливаем webpack

```
npm install webpack webpack-cli --save-dev
```

```
"dev": "webpack --mode development",  
"build": "webpack --mode production"
```

Используем entry/output

Переопределяем выходной файл бандла

Html webpack plugin

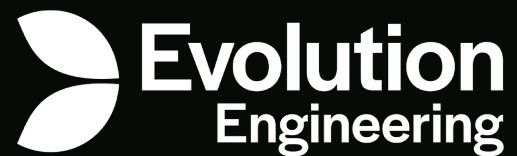
```
const HtmlWebpackPlugin = require('html-webpack-plugin');
```

```
plugins: [  
  new HtmlWebpackPlugin({  
    title: 'webpack Boilerplate',  
    template: path.resolve(__dirname, './src/template.html'), // шаблон  
    filename: 'index.html', // название выходного файла  
  }),  
],
```

src/template.html

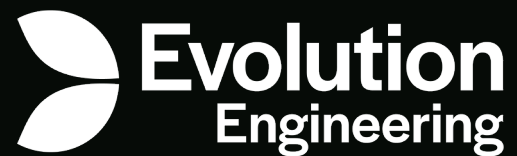
```
<!DOCTYPE html>
<html lang="en">
<head>
  <title><%= htmlWebpackPlugin.options.title %></title>
</head>

<body>
  <div id="root"></div>
</body>
</html>
```



```
const heading = document.createElement('h1');  
heading.textContent = 'Webpack lecture!';
```

```
const root = document.querySelector('#root');  
root.append(heading);
```



clean-webpack-plugin

Прибираемся за собой после пересборок

```
plugins: [  
  new HtmlWebpackPlugin( options: {  
    title: 'webpack Boilerplate',  
    template: path.resolve(__dirname, './src/template.html'), // шаблон  
    filename: 'index.html', // название выходного файла  
  }),  
  new CleanWebpackPlugin(),  
],
```

TODO: заменить объект
конфигурации на функцию с
аргументами env, argv

В зависимости от
`arg.mode("development", "production")`
изменить название output чанка

Production -> `[name][contenthash].bundle.js`
Development -> `[name].bundle.js`

```
> webpack --mode production
```

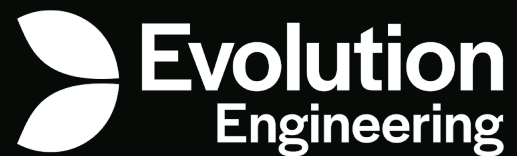
```
asset index.html 192 bytes [emitted]
```

```
asset main.f57badd23edcf6ee619a.bundle.js 122 bytes [emitted] [immutable] [minimized] (
```

```
./src/index.js 156 bytes [built] [code generated]
```

```
webpack 5.38.1 compiled successfully in 340 ms
```

```
aliakseikhatsianovich@Aliakseis-MacBook-Pro webpack-demo % npm run dev
```



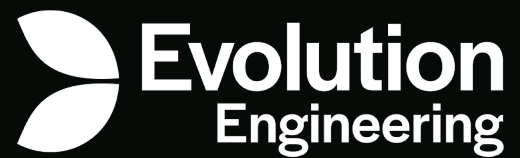
```
> webpack-demo@1.0.0 dev /Users/aliakseikhatsianovich/WebstormProjects/webpack-demo  
> webpack --mode development
```

```
asset main.bundle.js 1.34 KiB [emitted] (name: main)  
asset index.html 176 bytes [emitted]  
./src/index.js 156 bytes [built] [code generated]  
webpack 5.38.1 compiled successfully in 132 ms
```

Web-dev-server

```
npm install --save-dev webpack-dev-server
```

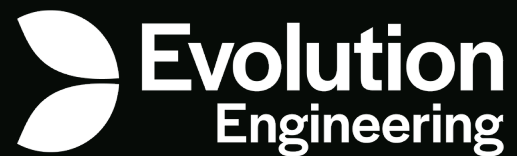
```
"dev": "webpack server --mode development",
```



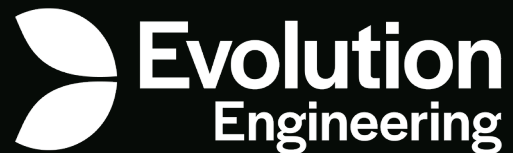
```
devServer: {  
  contentBase: './dist',  
  open: true  
},
```

Загрузки стилей(css)

```
npm install --save-dev css-loader style-loader
```



```
module: {  
  rules: [  
    {  
      test: /\.css$/,  
      use: ["style-loader", "css-loader"]  
    }  
  ],  
},
```



sass-loader — загружает SCSS и компилирует его в CSS

node-sass — Node Sass

postcss-loader — обработка CSS с помощью PostCSS

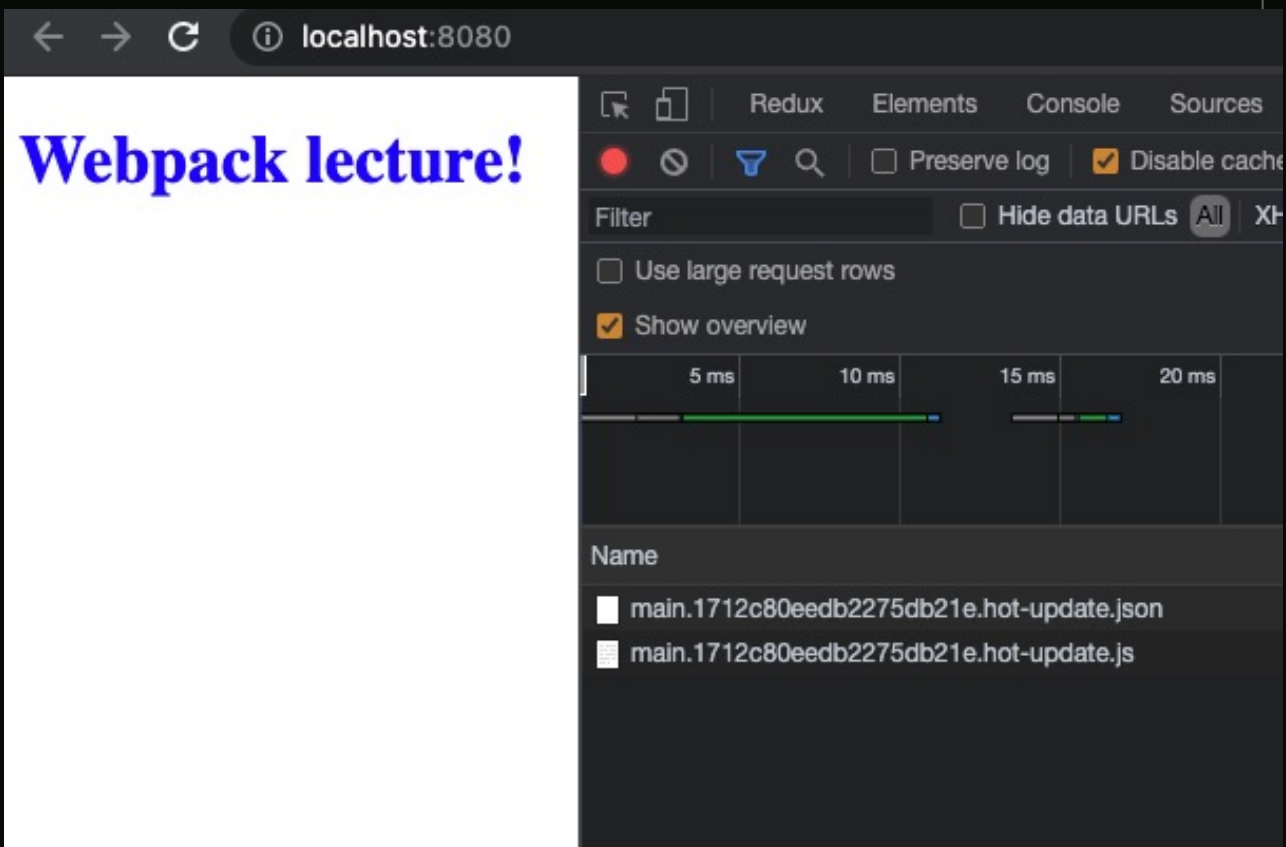
postcss-preset-env — полезные настройки PostCSS

css-loader — загрузка стилей

style-loader — применение стилей к элементам DOM

Hot module replacement

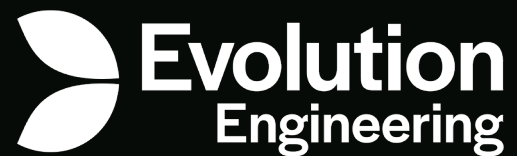
```
devServer: {  
  contentBase: './dist',  
  open: true,  
  hot: true  
},
```



The screenshot shows a web browser at `localhost:8080` displaying a slide titled "Webpack lecture!". To the right, the Chrome DevTools Performance tab is open, showing a timeline of events. The "Name" column lists two files: `main.1712c80eedb2275db21e.hot-update.json` and `main.1712c80eedb2275db21e.hot-update.js`. The timeline includes a filter bar with options like "Filter", "Hide data URLs", "All", "XHR", "Use large request rows", and "Show overview". A horizontal bar at the bottom indicates time intervals from 5 ms to 20 ms.

Загрузка svg

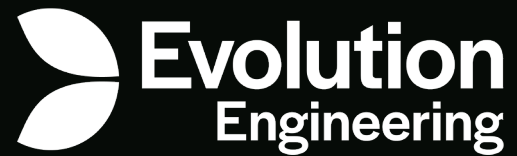
```
{  
  test: /\.svg$/,  
  type: 'asset/inline',  
},
```



```
import cat from './cat.svg'
```

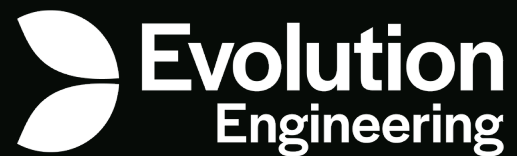
```
const catImg = document.createElement('img');  
catImg.src = cat;  
root.append(catImg);
```





Typescript

```
npm install --save-dev typescript ts-loader
```

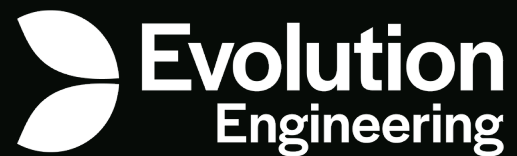


tsconfig.json

```
{  
  "compilerOptions": {  
    "outDir": "./dist/",  
    "noImplicitAny": true,  
    "module": "es6",  
    "target": "es5",  
    "jsx": "react",  
    "allowJs": true,  
    "moduleResolution": "node",  
  },  
  "include": ["types/custom.d.ts"]  
}
```

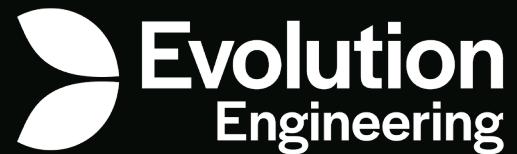
types/custom.d.ts

```
declare module "*.svg" {  
  const content: any;  
  export default content;  
}
```

webpack.config.js

```
resolve: {  
  extensions: ['.tsx', '.ts', '.js'],  
},  
  
{  
  test: /\.tsx?$/,  
  use: 'ts-loader',  
  exclude: /node_modules/,  
},
```



React

```
npm install --save-dev react react-dom @types/react @types/react-dom
```

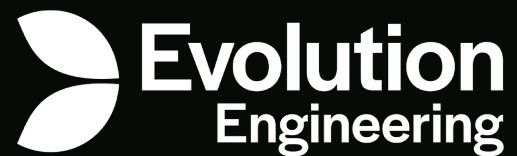
Сделать `src/index.tsx` приложение,
которое рендерит кнопку и по
клику увеличивает счетчик в `state`

```
import * as React from "react";
import { useState } from "react";
import { render } from "react-dom";

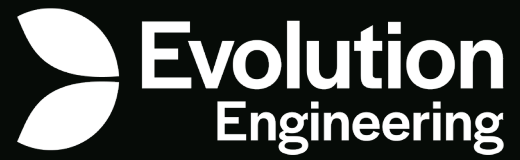
function App() {
  const [count, setCounter] = useState( initialState: 0 );

  return <button onClick={() => setCounter( value: count + 1 )}>`Clicked ${count}`</button>;
}

render(<App />, document.getElementById( elementId: "root" ));
```



Переименовать расширение в
конфиге точки входа ts->tsx

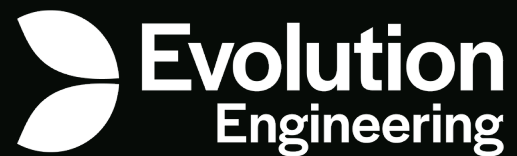


Code splitting

"build-dev": "webpack --mode development"

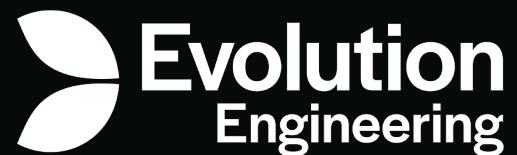
rootReact: ***path***.resolve(__dirname, './src/root.tsx'),
rootTs: ***path***.resolve(__dirname, './src/index.ts'),





Mini Css Extract Plugin

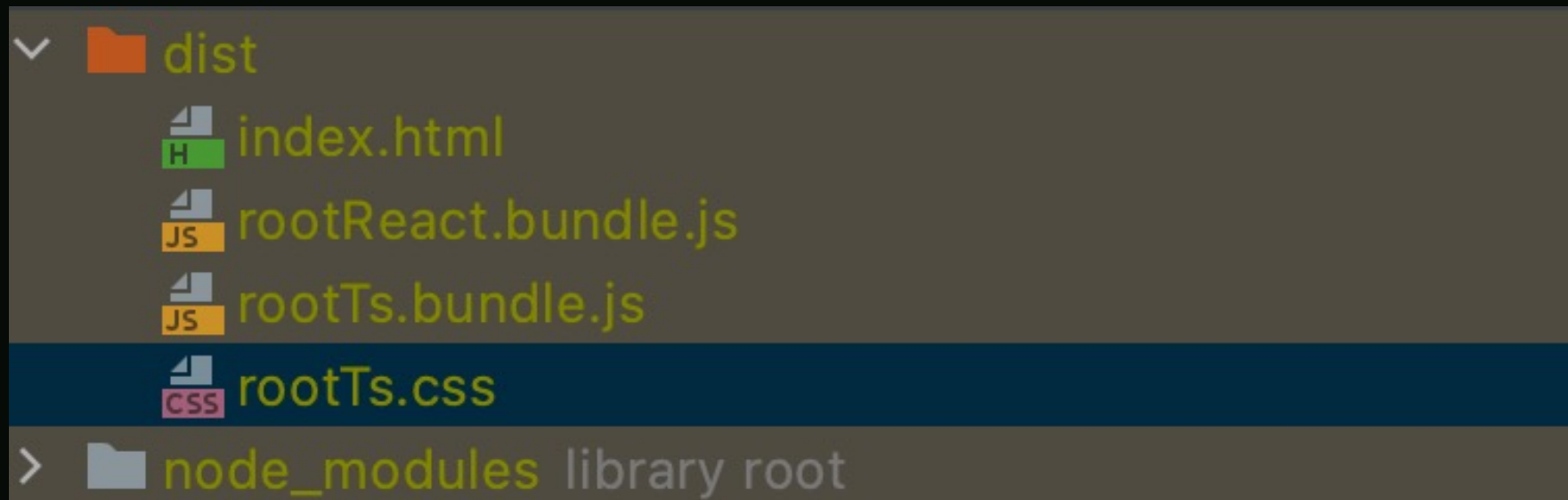
`npm install --save-dev mini-css-extract-plugin`

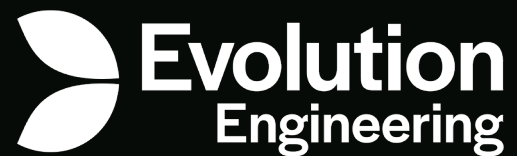


```
const MiniCssExtractPlugin = require('mini-css-extract-plugin');
```

```
use: [MiniCssExtractPlugin.loader, "css-loader"]
```

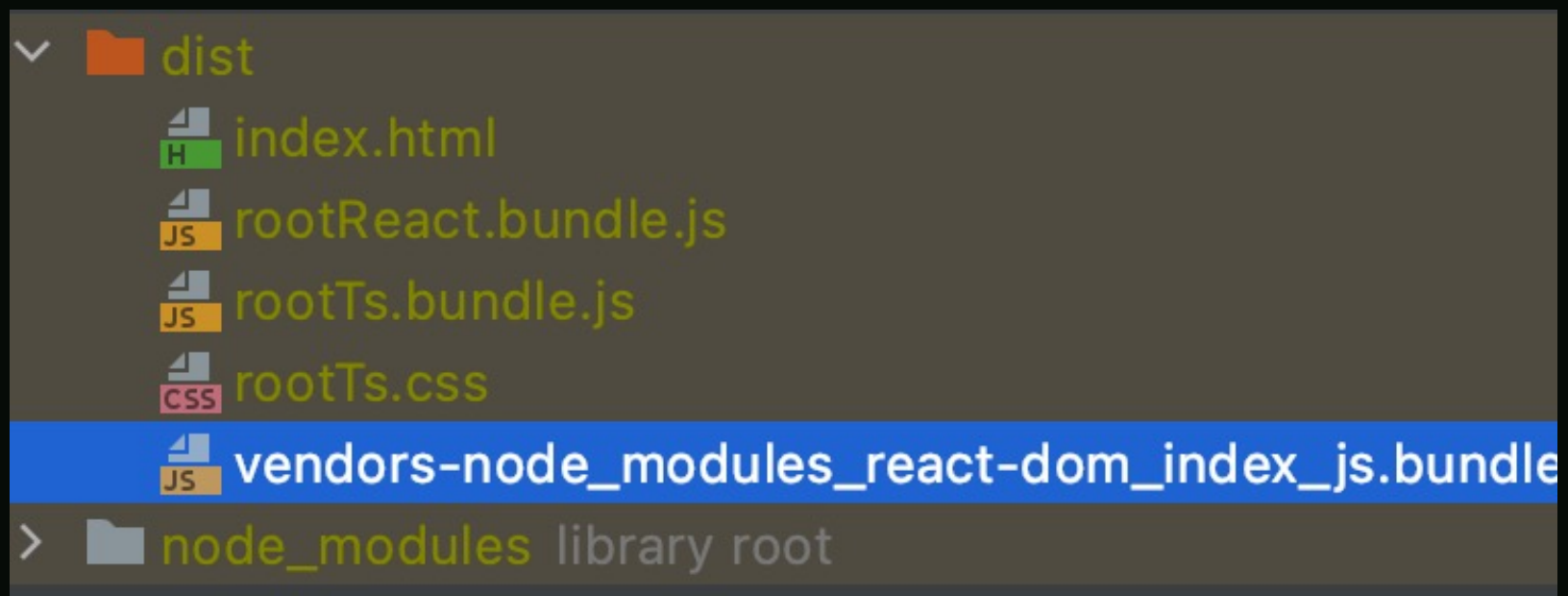
```
plugins: [  
  ...  
  new MiniCssExtractPlugin(),  
],
```





SplitChunksPlugin

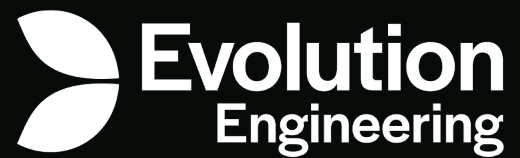
```
optimization: {  
  splitChunks: {  
    chunks: 'all',  
  },  
},
```



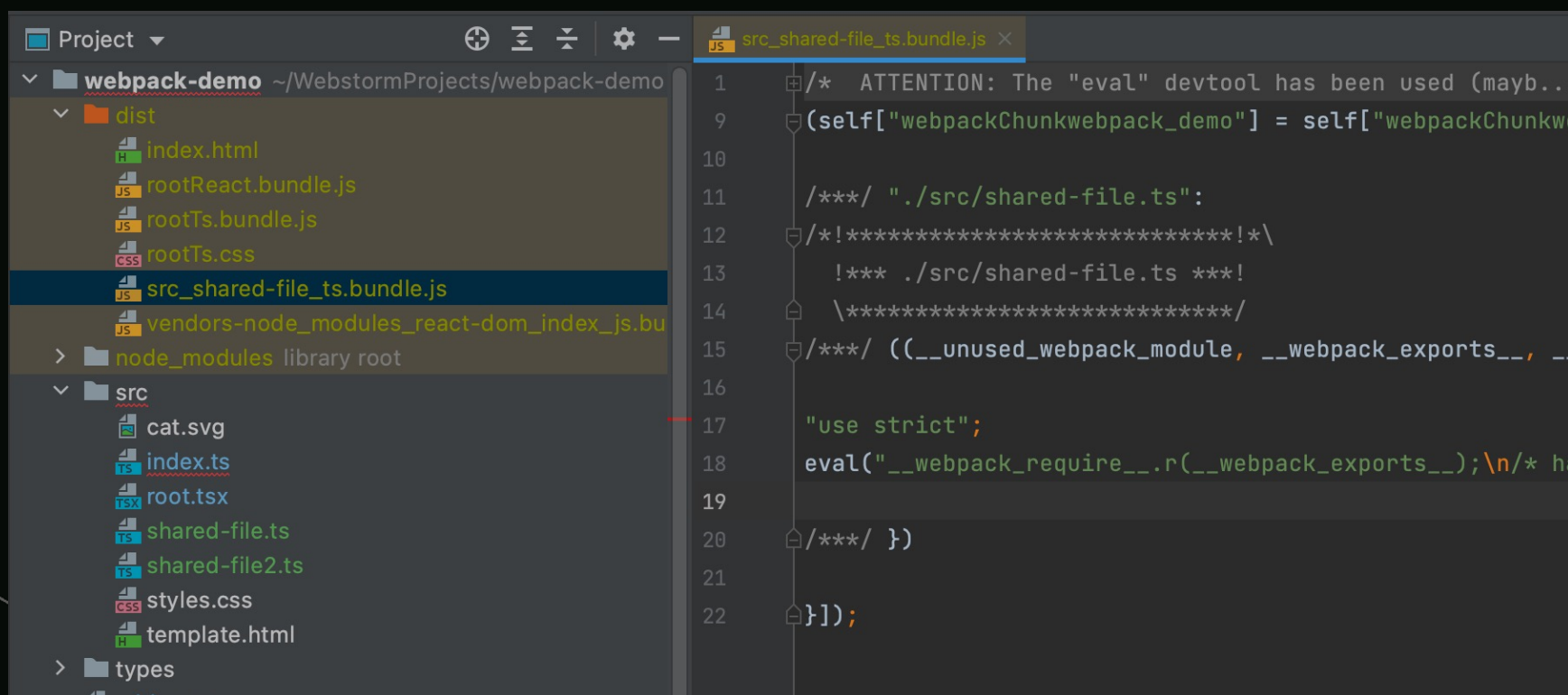
Проверка code splitting

```
import {logger} from "./shared-file";
```

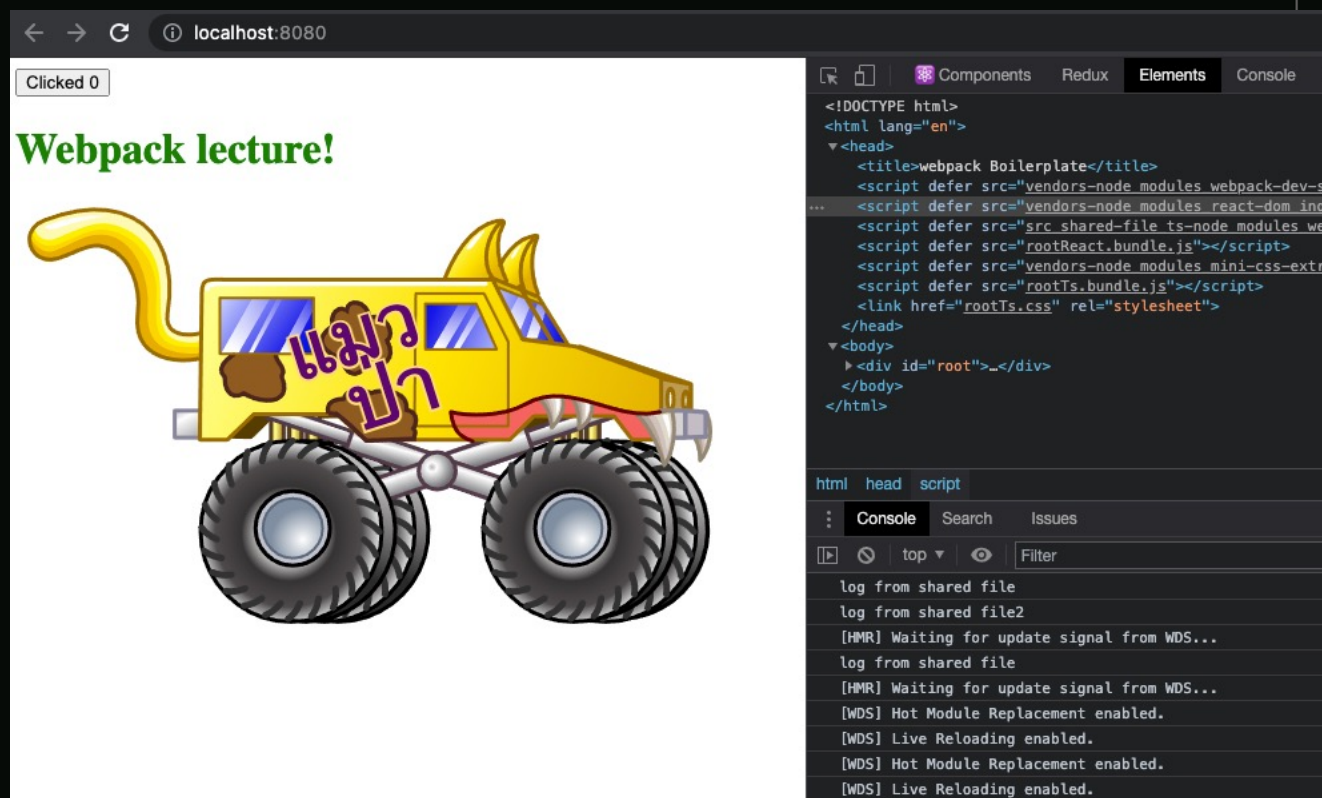
```
logger();
```

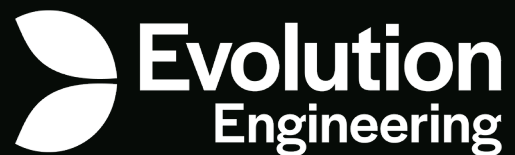


```
optimization: {  
  splitChunks: {  
    chunks: 'all',  
    minSize: 0,  
  },  
},
```

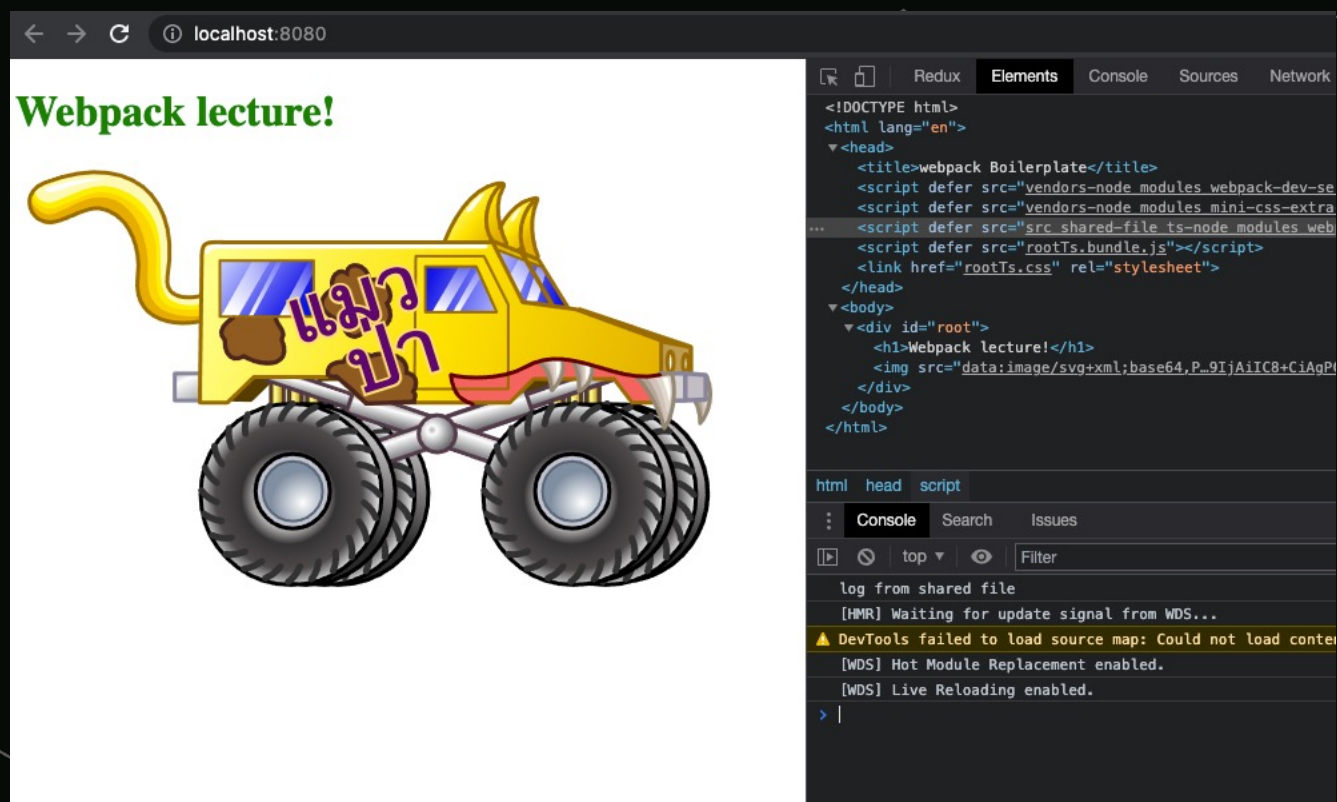


```
1  /* ATTENTION: The "eval" devtool has been used (mayb...
9  (self["webpackChunkwebpack_demo"] = self["webpackChunkwe
10
11  /**/ "./src/shared-file.ts":
12  /*!*****!*\
13  !*** ./src/shared-file.ts ***!
14  \*****/
15  /**/ ((__unused_webpack_module, __webpack_exports__, __
16
17  "use strict";
18  eval("__webpack_require__\.r(__webpack_exports__); \n/* ha
19
20  /**/ })
21
22  {}]);
```



```
new HtmlWebpackPlugin({  
  title: 'webpack Boilerplate',  
  template: path.resolve(__dirname,  
    './src/template.html'), // шаблон  
  filename: 'index.html', // название  
    выходного файла  
  chunks : ['rootTs'],  
}),
```



Динамический импорт

```
setTimeout(()=>{  
  import("./heavy-file");  
}, 3000);
```

Webpack lecture!



Redux Elements Console Sources **Network** Performance Memory Application Security Lighthouse Adblock Plus

Filter ☐ Hide data URLs ☒ All XHR JS CSS Img Media Font Doc WS Manifest Other ☐ Has blocked cookies ☐ Blocked Requests

☐ Use large request rows ☐ Group by frame

☒ Show overview ☐ Capture screenshots

Name	Status	Type	Initiator	Size	Time	Waterfall
localhost	200	document	Other	821 B	2 ms	
vendors-node_modules_webpack-dev-server_client_ind...localhost_8080-node_mod...	200	script	(index)	376 kB	8 ms	
vendors-node_modules_mini-css-extract-plugin_dist_hmr_hotModuleReplacement_js...	200	script	(index)	7.6 kB	8 ms	
rootTs.css	200	stylesheet	(index)	274 B	6 ms	
src_shared-file_ts-node_modules_webpack_hot_sync_log_bundle.js	200	script	(index)	2.7 kB	2 ms	
rootTs.bundle.js	200	script	(index)	198 kB	3 ms	
data:image/svg+xml;...	200	svg+xml	index.ts:14	(memory cache)	1 ms	
info?t=1622805672875	200	xhr	VM839:1	391 B	2 ms	
websocket	101	websocket	sockjs.js:1687	0 B	Pe...	
error_38.png	200	png	content.js:36	1.5 kB	2 ms	
src_heavy-file_ts.bundle.js	200	script	rootTs.bundle.js:273	1.2 kB	2 ms	

http://localhost:8080/src_heavy-file_ts.bundle.js

Полезные ссылки:

- <https://webpack.js.org/concepts/>
- <https://github.com/khats/webpack-demo>