


home > jepscrk > Downloads > webpack-boilerplate >  webpack.config.js > ...

```
1  const path = require('path'); // CommonJS
2
3  module.exports = {
4    mode: 'production',
5    entry: './src/index.js',
6    output: {
7      path: path.resolve(__dirname, 'public', 'assets', 'js'),
8      filename: 'bundle.js'
9    },
10   module: {
11     rules: [{
12       exclude: /node_modules/,
13       test: /\.js$/,
14       use: {
15         loader: 'babel-loader',
16         options: {
17           presets: ['@babel/env']
18         }
19       }
20     }]
21   },
22   devtool: 'source-map'
23 };
```

Explicação do webpack.config.js

1. `const path = require('path');`

Importa o módulo 'path' do Node.js. Esse módulo ajuda a montar caminhos de diretórios de forma segura entre diferentes sistemas operacionais.

2. `module.exports = { mode: 'production',`

Define que o Webpack vai gerar o código otimizado para produção. Isso significa código minificado e mais leve.

3. `entry: './src/index.js',`

Define o ponto de entrada do seu aplicativo. Webpack começa a analisar o projeto a partir desse arquivo.

4. `output: { path: path.resolve(__dirname, 'public', 'assets', 'js'), filename: 'bundle.js' }`

Diz onde será salvo o arquivo final (bundle). Será gerado um bundle.js na pasta public/assets/js.

5. `module: { rules: [...] }`

Define que arquivos .js (exceto node_modules) devem ser processados com Babel, convertendo código moderno em código compatível com navegadores antigos.

```
👉 Aqui você define que o Webpack deve **usar o Babel** pra transformar seu código JS

- `test: /\.js$/` → Aplica isso a todos os arquivos `.js`.
- `exclude: /node_modules/` → Ignora libs externas.
- `loader: 'babel-loader'` → Usa o Babel.
- `presets: ['@babel/env']` → Converte seu JS moderno (ES6+) pra algo mais compatível

---

### ```js
  devtool: 'source-map'
```

6. devtool: 'source-map'

Gera um arquivo .map para permitir debug do código original, mesmo após minificação.

- Entrada: src/index.js
- Transpilação com Babel
- Bundle gerado em: public/assets/js/bundle.js
- Código otimizado para produção
- Compatibilidade com navegadores antigos
- Suporte a debug via source map