

## 38 - Utilizando o webpack-dev-server

1. Para essa aula iremos utilizar o projeto do último exercício, assim já teremos a estrutura principal pronta. Você pode seguir os passos na mesma pasta do exercício ou criar uma nova pasta e copiar o conteúdo para ela.
2. Com tudo pronto, a primeira coisa a fazer para utilizar o webpack-dev-server é instalar o pacote através do comando:

```
npm i -D webpack-dev-server
```

3. O webpack-dev-server se integra diretamente ao webpack, então tudo que precisamos fazer para configurá-lo é editar as opções do mesmo arquivo "webpack.config.js" que já tínhamos:

```
// webpack.config.js const path = require('path') module.exports = {  
  devServer: { static: { directory: path.resolve(__dirname, 'dist') },  
  compress: true, port: 8000 }, entry: { index: './src/index.js' },  
  mode: 'production', module: { rules: [{ test: /\.css$/, use: ['style-  
  loader', 'css-loader'] }, { test: /\.js$/, use: ['babel-loader'] }] },  
  output: { filename: '[name].min.js' } }
```

4. Com tudo configurado, já podemos executar o webpack-dev-server. Mas, por uma questão de conveniência, podemos criar um script para ele no arquivo "package.json":

```
// package.json // ... "scripts": { "build": "webpack", "dev":  
  "webpack-dev-server" }, // ...
```

## 5. E então executar o script:

Obs.: Repare que agora temos todos os benefícios do webpack unidos ao webpack-dev-server, o que garante:

- Reempacotamento automático sempre que uma alteração for salva
- Projeto sendo servido na porta configurada (sem precisar da extensão Live Server)
- Página do navegador recarregada sempre uma alteração for salva
- E outras opções que podem ser conferidas na documentação

```
npm run dev
```