

Lab 3. Webpack merge

In this lab you will use different setups to split up the configuration into multiple settings

duration: 30 minutes

prerequisites

You need to have a package.json file in the root of the target folder and you need to have webpack installed globally for these labs to succeed.

Step 1. Install the requirements

Navigate into the earlier created directory 'webpack_labs'.
Install webpack-merge using

```
$ npm install webpack-merge
```

create a file 'test.js' inside your current folder and write the following code, only if it does not already exists.

```
console.log('test');
```

Save the results.

Step 1. Use a single file merge

Navigate into the earlier created directory 'webpack_labs' and create or open the file 'webpack.config.js' inside your current folder. Write the following code into this file:

```

var webpack = require('webpack')
var merge = require('webpack-merge')

var commonconfig = {
  context: __dirname,
  entry: "./test.js",
  output: {
    filename: "bundle.js"
  }
}

var PROD = (process.env.NODE_ENV === 'production')
var config = {
  output: {
    path: __dirname + "/dev"
  }
};
if (PROD){
  config = {
    output: {
      path: __dirname + "/prod",
      filename: "bundle.min.js"
    },
    plugins: [
      new webpack.optimize.UglifyJsPlugin({minimize: true})
    ]
  }
}

module.exports = merge(commonconfig, config);

```

Use webpack to generate the bundles:

```
webpack
```

```
export NODE_ENV=production
webpack
```

Inspect the bundles in the folders 'dev' and 'prod', note how the production version is a minified version of the bundle, whilst the dev version of the bundle is a readable, commented version of the bundle.