

## Lab 2. Basic Setup

In this lab you will extract and use webpack configuration files

duration: 30 minutes

### Step 1. Create a new webpack.config.js

Navigate into the earlier created directory 'webpack\_labs' and create a file 'webpack.config.js' inside your current folder. Write the following code into this file:

```
module.exports = {
  context: __dirname,
  entry: "./test.js",
  output: {
    path: __dirname + "/dist",
    filename: "bundle.js"
  }
}
```

Save the file.

Run the command to build the bundle.js. Use your favorite text editor to inspect the generated file. Notice that we used an output directory with the name 'dist' this time to generate our javascripts.

Run the file using the following command:

```
$ node ./dist/bundle.js
```

If all went well, the output should display

```
test
this second file is bundled aswell
```

### Step 2. Rename the config and use the command line

Add another file, named webpack.test.js and fill it with the following content:

```
module.exports = {
  context: __dirname,
  entry: "./test.js",
  output: {
    path: __dirname + "/build",
    filename: "bundle.js"
  }
}
```

Note how the output directory is specific set to 'build'.

Save the file and rebuild the bundle with the following command:

```
$ webpack --config webpack.test.js
```

Note how the 'build' directory is generated and the bundle is in there aswell. Test the workings of the code by running the bundle.

### **Step 3. Extra exercise**

Use your knowlegde and add more modules that are circular referenced to each other, so a.js requires b.js and b.js requires c.js, which in turn requires a.js. Try to use webpack to create the desired bundle and inspect the results.