

# Getting Started with Node.js and NPM



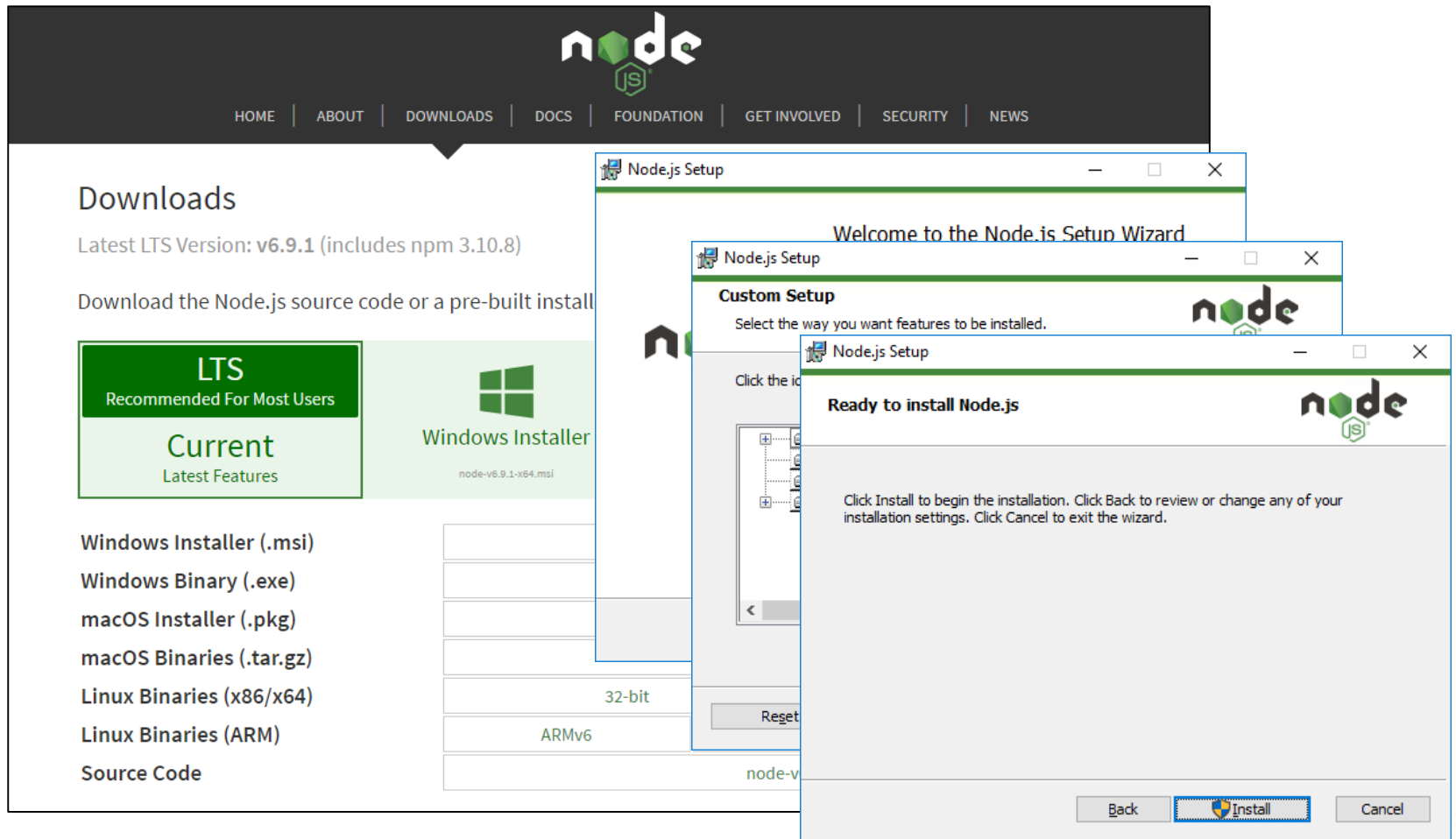
# Agenda

- Introduction to Node.JS and Visual Studio Code
  - Installing and Updating NPM packages
  - Configuring Server-side Debugging Support
  - Node.JS Development with TypeScript
  - Using Gulp to Automate Running Tasks
  - Bundling the Source Files using WebPack



# Installing node.js

- <https://nodejs.org/en/download/>



# Cross-platform Toolchain

- Node.js
- Node Package Manager (npm)
- TypeScript
- Yeoman
- Webpack
- Gulp
- git

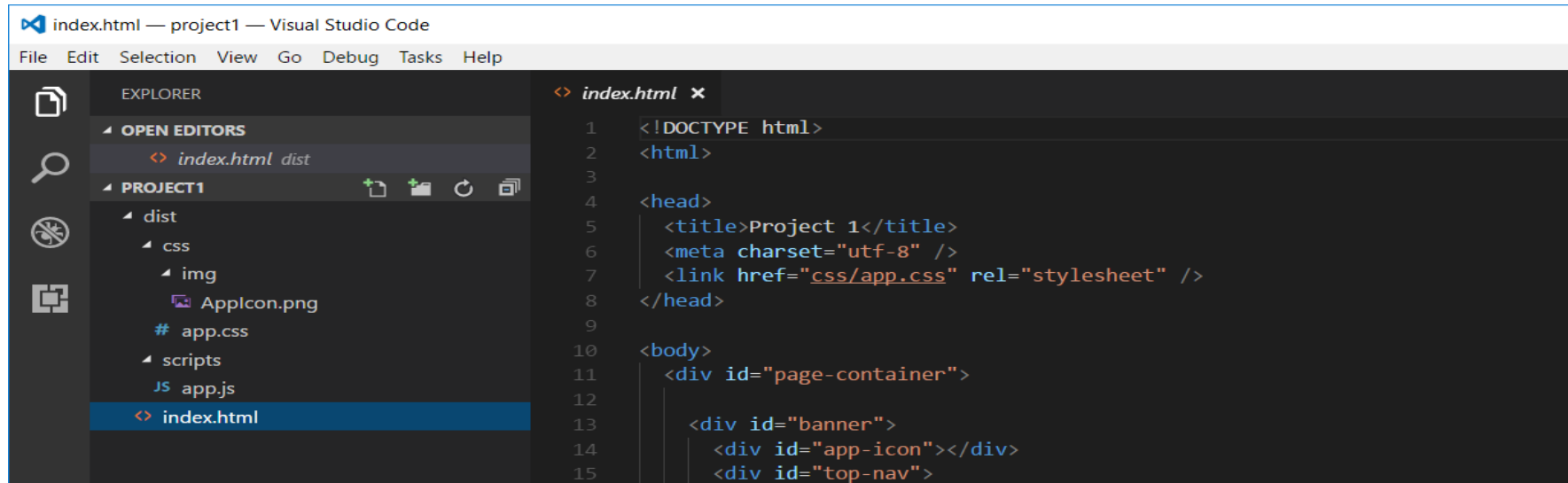


# Install Visual Studio Code

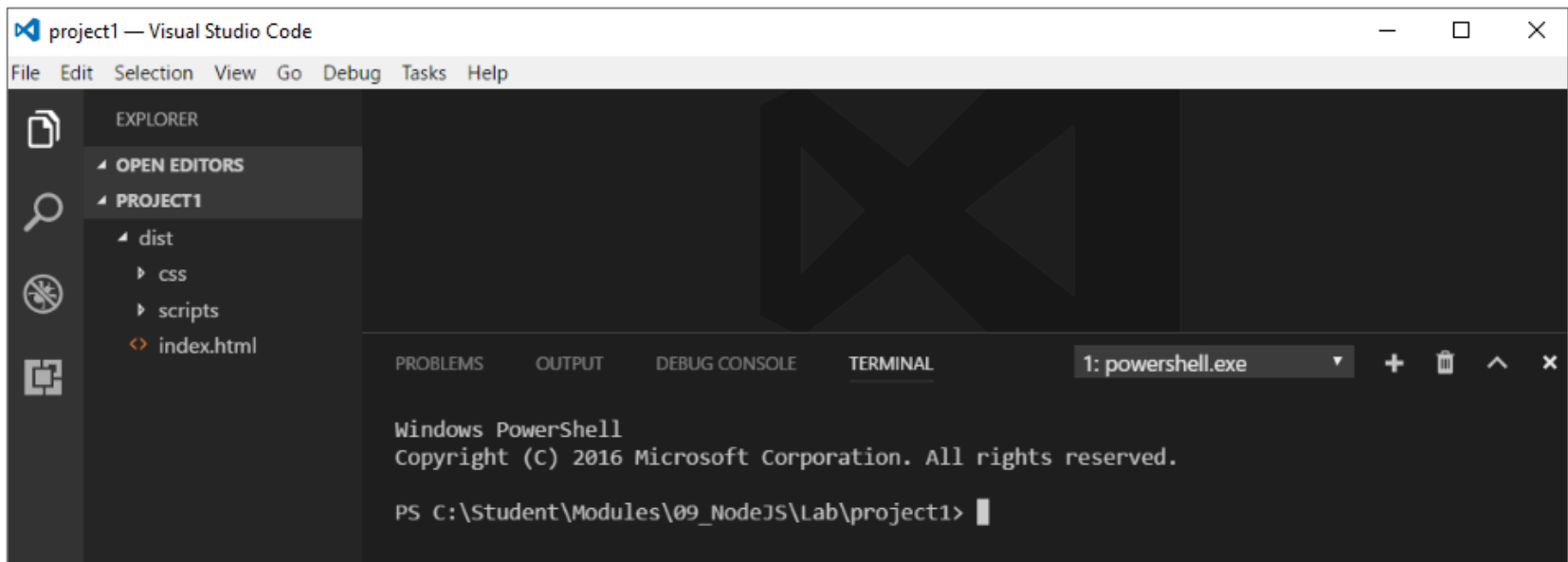
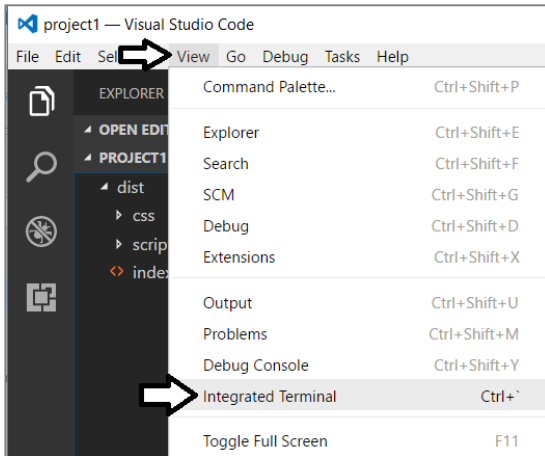
- <http://code.visualstudio.com/>



# Developing with Visual Studio Code



# Integrated Terminal





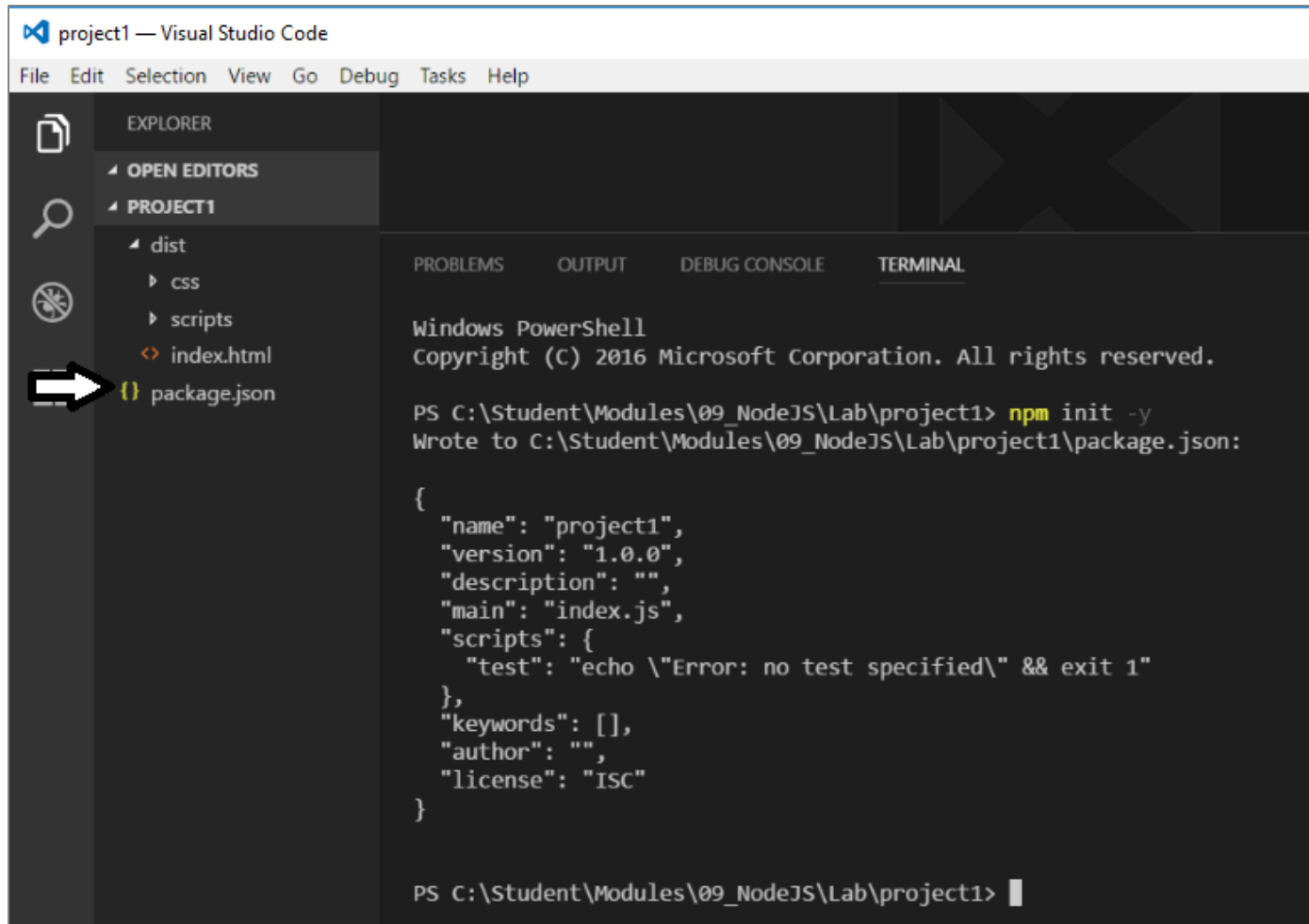
# Agenda

- ✓ Introduction to Node.JS and Visual Studio Code
- Installing and Updating NPM packages
  - Configuring Server-side Debugging Support
  - Node.JS Development with TypeScript
  - Using Gulp to Automate Running Tasks
  - Bundling the Source Files using WebPack





# npm init



project1 — Visual Studio Code

File Edit Selection View Go Debug Tasks Help

EXPLORER

- OPEN EDITORS
- PROJECT1
  - dist
    - css
    - scripts
    - index.html
    - package.json

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Windows PowerShell  
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

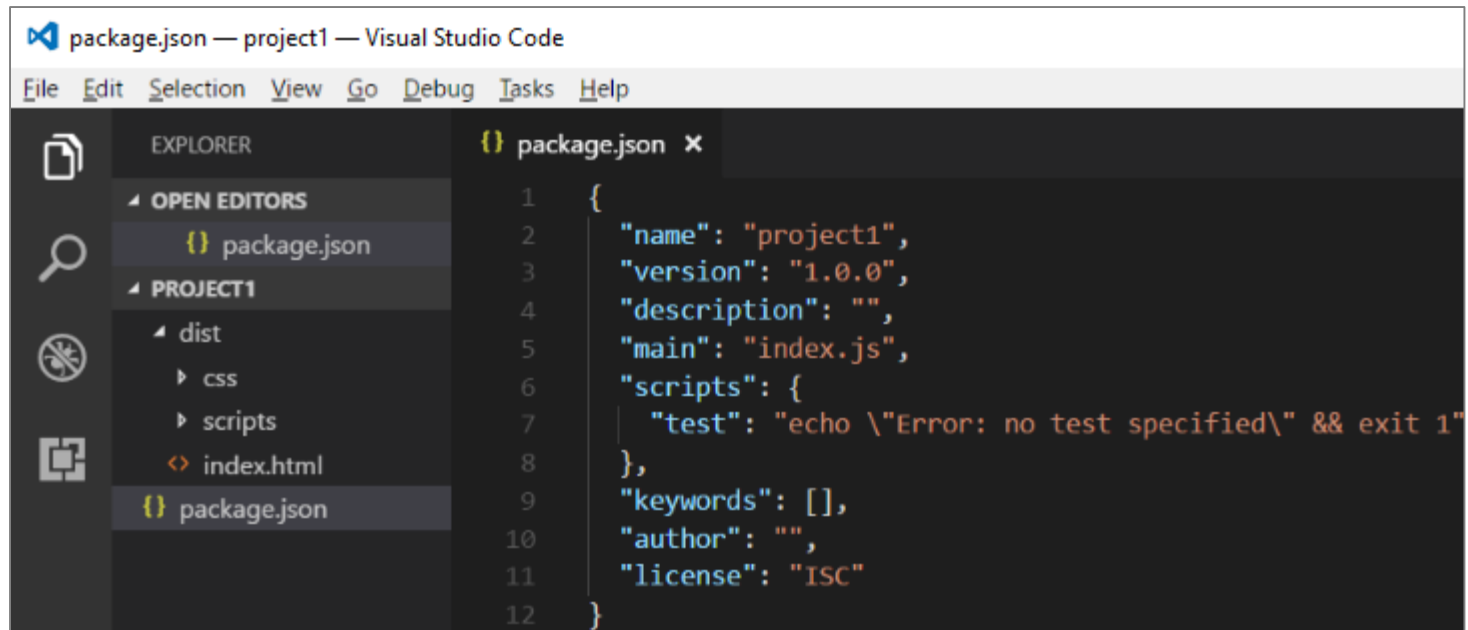
PS C:\Student\Modules\09\_NodeJS\Lab\project1> npm init -y  
Wrote to C:\Student\Modules\09\_NodeJS\Lab\project1\package.json:

```
{
  "name": "project1",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
```

PS C:\Student\Modules\09\_NodeJS\Lab\project1> |



# package.json



The screenshot shows the Visual Studio Code interface with the `package.json` file open. The Explorer view on the left shows the project structure, including the `package.json` file. The Editor view on the right shows the content of the `package.json` file, which is a JSON object with the following properties:

```
1 {  
2   "name": "project1",  
3   "version": "1.0.0",  
4   "description": "",  
5   "main": "index.js",  
6   "scripts": {  
7     "test": "echo \"Error: no test specified\" && exit 1"  
8   },  
9   "keywords": [],  
10  "author": "",  
11  "license": "ISC"  
12 }
```



# Installing Packages

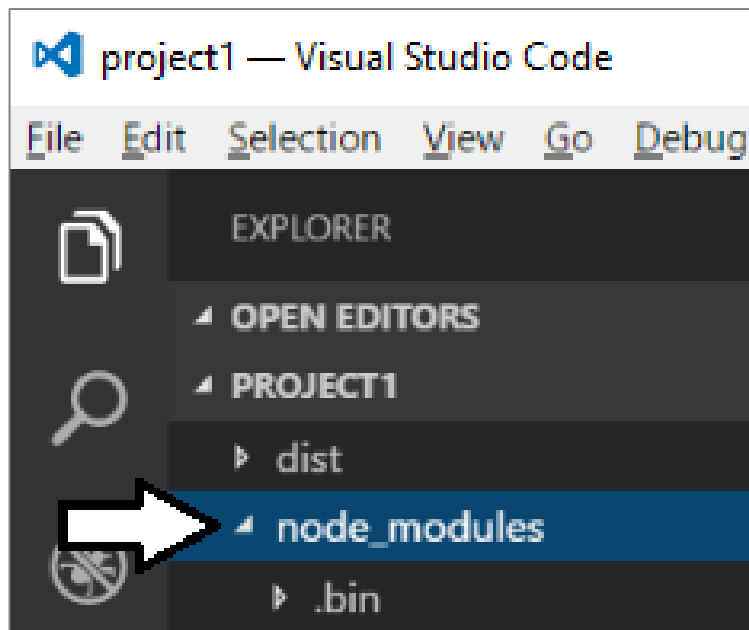
```
npm install browser-sync --save-dev
```

```
"devDependencies": {  
  "browser-sync": "^2.18.12"  
}
```



# node\_modules folder

- Package files copied into **node\_modules** folder
  - This folder often contain 100s of packages for a project
  - Contents of folder not saved into source control
  - Contents can be restored with **npm install** command



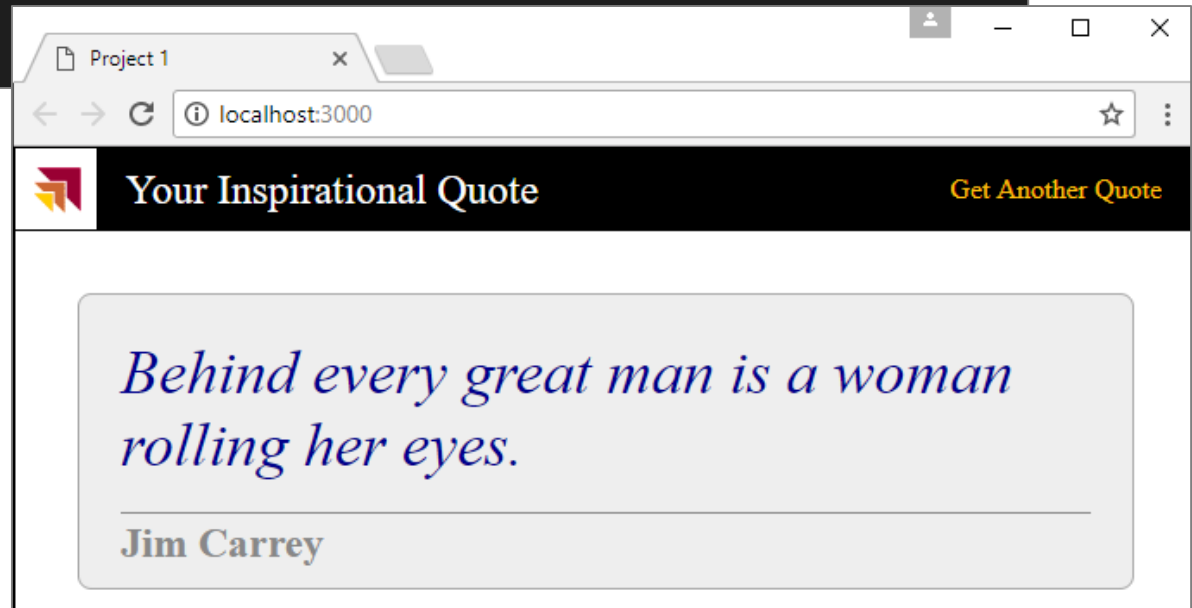
# Agenda

- ✓ Introduction to Node.JS and Visual Studio Code
- ✓ Installing and Updating NPM packages
- Configuring Server-side Debugging Support
  - Node.JS Development with TypeScript
  - Using Gulp to Automate Running Tasks
  - Bundling the Source Files using WebPack



# Using browser-sync to Serve Content

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    1: powershell.exe +
PS C:\Student\Modules\09_NodeJS\Lab\project1> browser-sync start --server dist
[BS] Access URLs:
-----
Local: http://localhost:3000
External: http://192.168.1.196:3000
-----
UI: http://localhost:3001
UI External: http://192.168.1.196:3001
-----
[BS] Serving files from: dist
```



# Stopping the Web Server Session

- Type CTRL + C into console to interrupt session


```
-----  
Local: http://localhost:3000  
External: http://192.168.1.196:3000  
-----  
UI: http://localhost:3001  
UI External: http://192.168.1.196:3001  
-----  
[BS] Serving files from: dist  
^CTerminate batch job (Y/N)? █
```





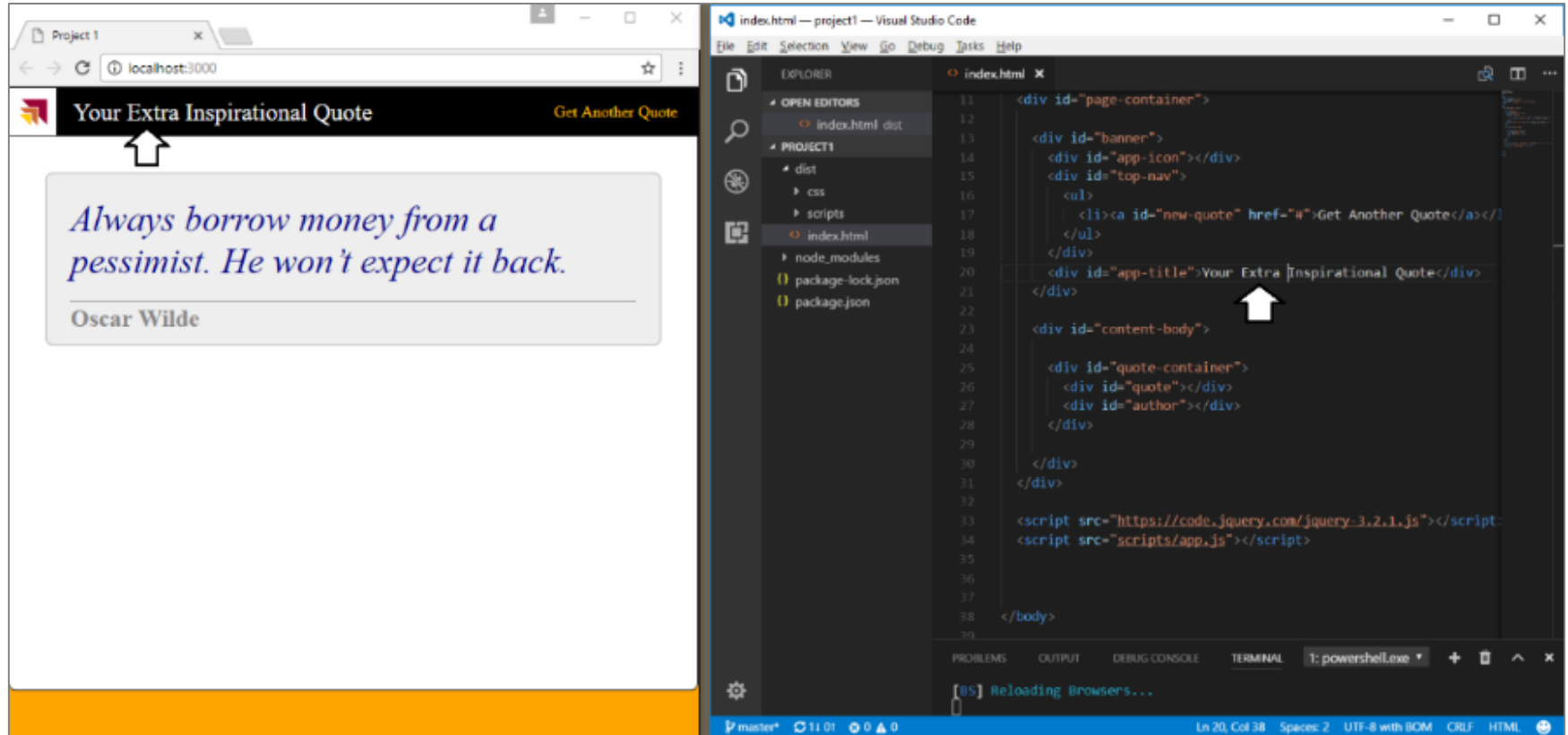
# Watching Files

- Issue this command
  - `browser-sync start --server dist --files dist`

```
PS C:\Student\Modules\09_NodeJS\Lab\project1> browser-sync start --server dist --files dist
rver dist --files dist
[BS] Access URLs:
-----
    Local: http://localhost:3000
  External: http://192.168.1.196:3000
-----
    UI: http://localhost:3001
  UI External: http://192.168.1.196:3001
-----
[BS] Serving files from: dist
[BS] Watching files... 
```



# Automatic Updates



# Agenda

- ✓ Introduction to Node.JS and Visual Studio Code
- ✓ Installing and Updating NPM packages
- ✓ Configuring Server-side Debugging Support
- Node.JS Development with TypeScript
  - Using Gulp to Automate Running Tasks
  - Bundling the Source Files using WebPack



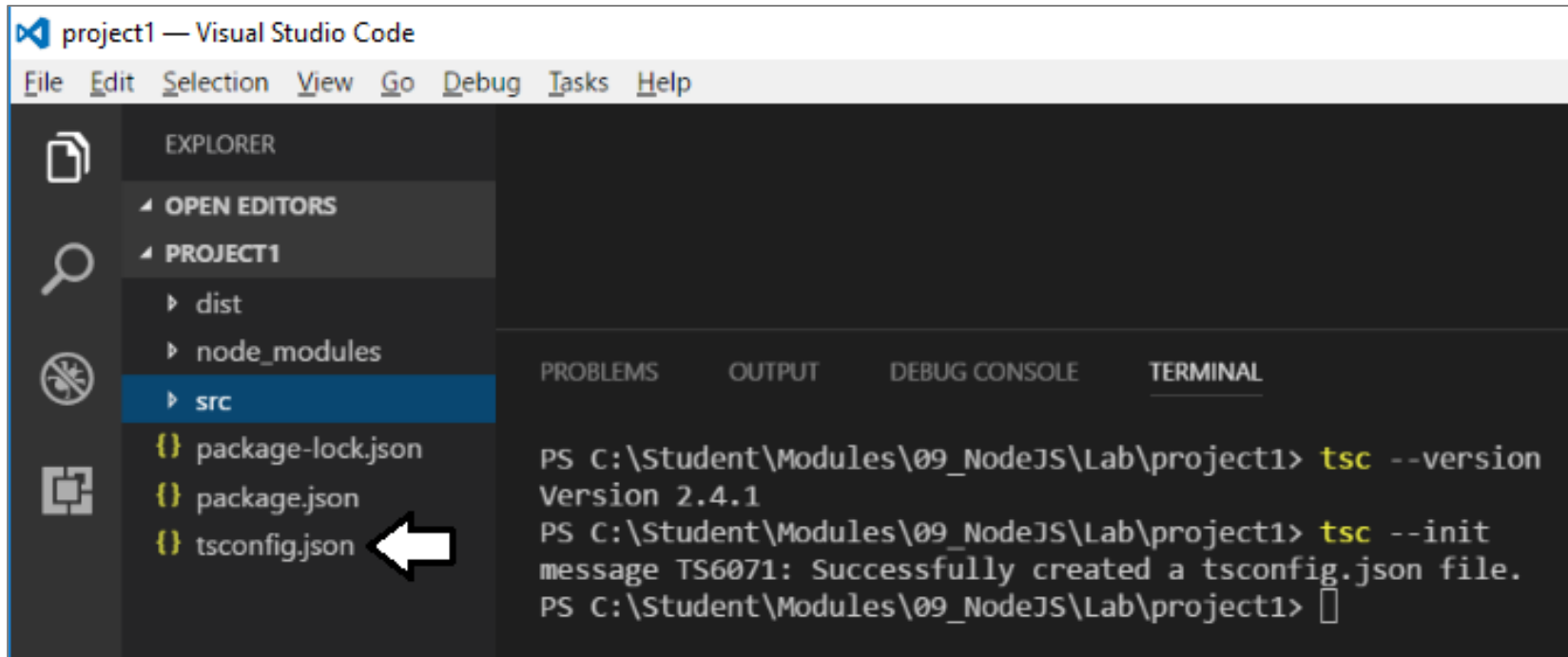
# Installing the TypeScript Package

```
npm install typescript --save-dev
```

```
PS C:\Student\Modules\09_NodeJS\Lab\project1> tsc --version  
Version 2.4.1  
PS C:\Student\Modules\09_NodeJS\Lab\project1> █
```



# Generating tsconfig.json

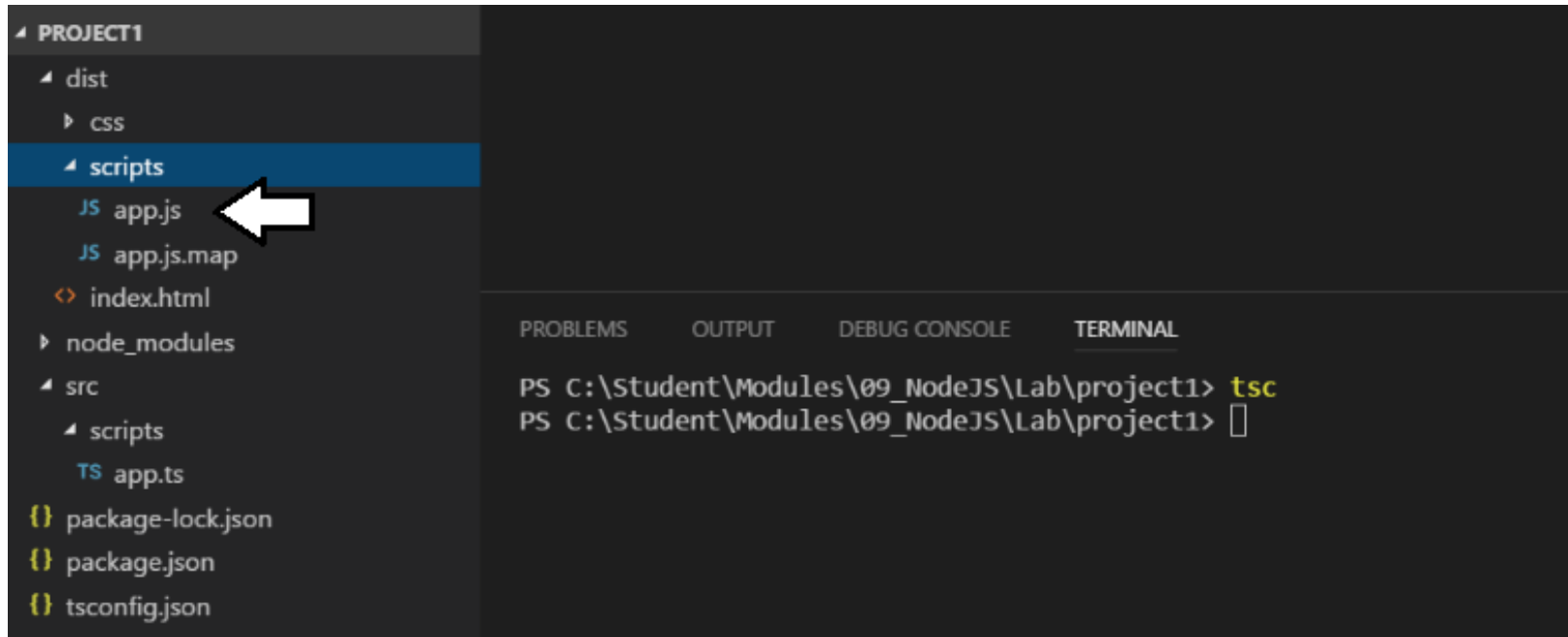


# tsconfig.json

```
{
  "compilerOptions": {
    "noImplicitAny": true,
    "removeComments": true,
    "preserveConstEnums": true,
    "outFile": "./dist/scripts/app.js",
    "sourceMap": true,
    "lib": [ "dom", "es6" ]
  },
  "files": [
    "./src/scripts/app.ts"
  ],
  "exclude": [
    "node_modules"
  ]
}
```

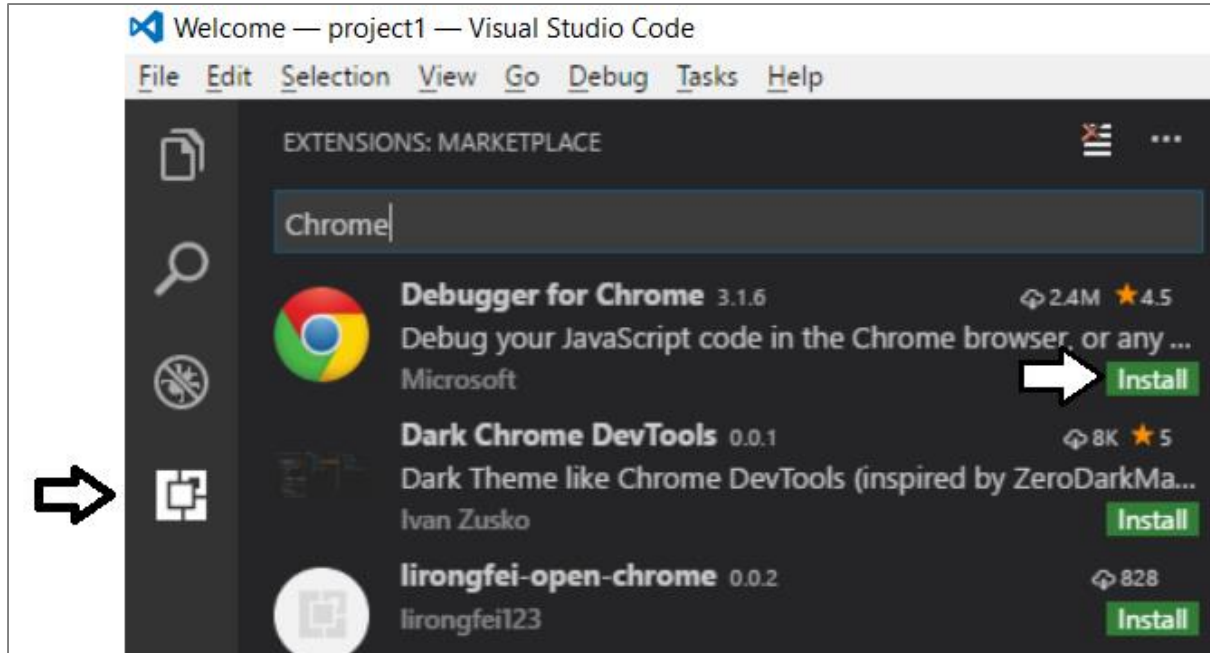


# Running tsc





# Chrome Debugging Support



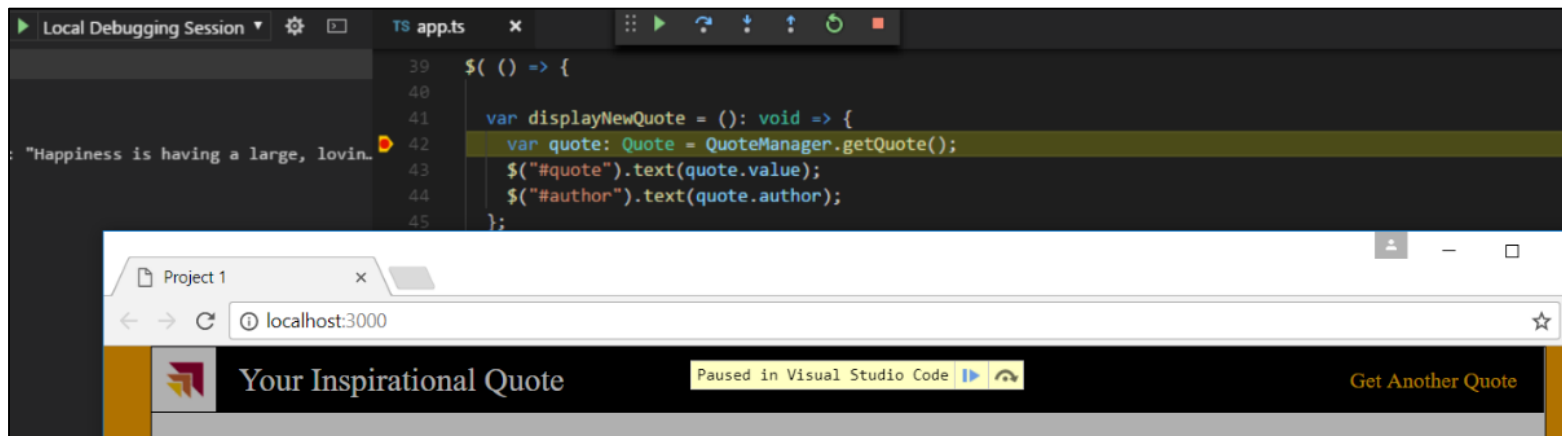
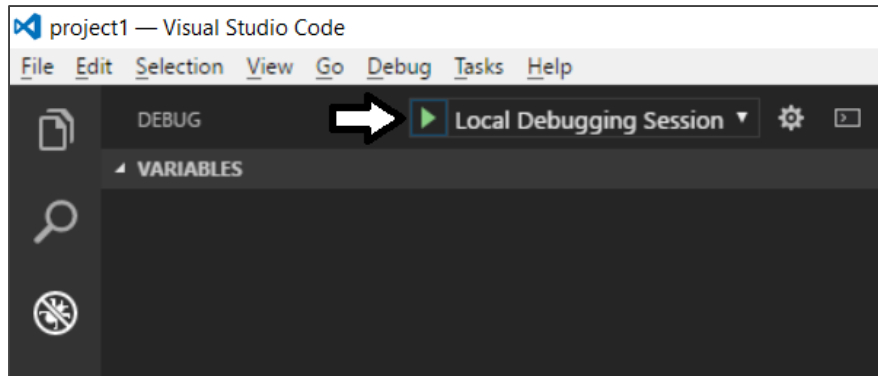
# Visual Studio Debugging Support

- Debugging configurations tracked in launch.json

```
{ } launch.json ●
1  {
2      "version": "0.2.0",
3      "configurations": [
4          {
5              "name": "Local Debugging Session",
6              "type": "chrome",
7              "request": "launch",
8              "url": "http://localhost:3000/",
9              "webRoot": "${workspaceRoot}/dist",
10             "sourceMaps": true,
11             "runtimeArgs": [
12                 "--remote-debugging-port=9222"
13             ]
14         }
15     ]
16 }
```



# Running the Debugger



# Agenda

- ✓ Introduction to Node.JS and Visual Studio Code
- ✓ Installing and Updating NPM packages
- ✓ Configuring Server-side Debugging Support
- ✓ Node.JS Development with TypeScript
- Using Gulp to Automate Running Tasks
- Bundling the Source Files using WebPack



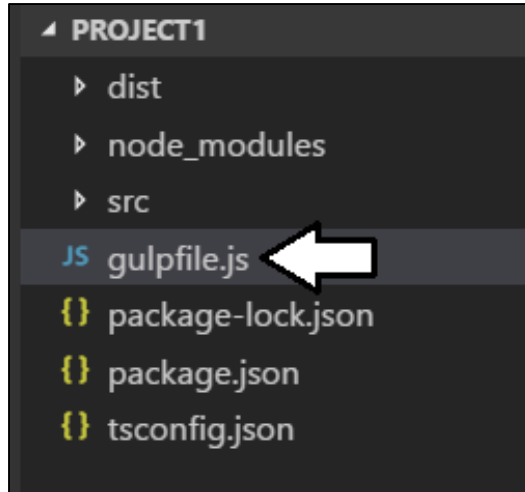
# Gulp as a Task Runner

- Gulp serves as a Task Runner
  - Compiles TypeScript files to JavaScript
  - Compiles SASS files to CSS
  - Bundles and minifies JavaScript and CSS files

```
npm install gulp --save-dev
```



# gulpfile.js



```
var gulp = require('gulp');

gulp.task('default', function() {
  console.log("Running my very first gulp task")
});
```



# Agenda

- ✓ Introduction to Node.JS and Visual Studio Code
- ✓ Installing and Updating NPM packages
- ✓ Configuring Server-side Debugging Support
- ✓ Node.JS Development with TypeScript
- ✓ Using Gulp to Automate Running Tasks
- Bundling the Source Files using WebPack





# WebPack

- WebPack serves as a bundling utility
  - Bundles many js/ts files into a single file
  - Can handle dynamic module loading
  - Provides a dev server for testing and debugging



# Dynamic Module Loading

```
TS app.ts x
import { Quote } from './quote';
import { QuoteManager } from './quote-manager';

$( () => {
  var displayNewQuote = (): void => {
    var quote: Quote = QuoteManager.getQuote();
    $("#quote").text(quote.value);
    $("#author").text(quote.author);
  }
});
```

```
TS quote.ts •
1 export class Quote {
2   value: string;
3   author: string;
4   constructor(value: string, author: string) {
5     this.value = value;
6     this.author = author;
7   }
8 }
```

```
TS quote-manager.ts x
1 import { Quote } from './quote';
2
3 export class QuoteManager {
4
5   private static quotes: Quote[] = [
6     new Quote("Always borrow money from a partner", "John D. Rockefeller"),
7     new Quote("Behind every great man is a great woman", "Florence Nightingale"),
8     new Quote("In Hollywood a marriage is a business", "Marilyn Monroe")
9   ];
10 }
```



# webpack.config.js

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

module.exports = {
  entry: {
    'app': './src/scripts/app.ts'
  },
  output: {
    path: __dirname + '/dist',
    filename: './scripts/bundle.js'
  },
  resolve: {
    extensions: ['.ts', '.js']
  },
  module: {
    rules: [{
      test: /\.ts$/,
      use: 'ts-loader'
    }]
  }
};
```

## PROJECT2

▸ node\_modules

▸ src

{ } package-lock.json

{ } package.json

{ } tsconfig.json

JS webpack.config.js



# Webpack Builds

```
PS C:\Student\Modules\09_NodeJS\Lab\project2> npm run build
```

```
> project2@1.0.0 build C:\Student\Modules\09_NodeJS\Lab\project2
> webpack
```

```
ts-loader: Using typescript@2.4.2 and C:\Student\Modules\09_NodeJS\Lab\project2\tsconfig.json
```

```
Hash: 2620a7c45f47a0cc754e
```

```
Version: webpack 2.7.0
```

```
Time: 1573ms
```

	Asset	Size	Chunks		Chunk Names
	./scripts/bundle.js	3.43 kB	0 [emitted]		app
	./scripts/vendor.bundle.js	275 kB	1 [emitted] [big]		vendor
[0]	./~/jquery/dist/jquery.js	268 kB	{1} [built]		
[1]	./src/scripts/quote-manager.ts	2.42 kB	{0} [built]		
[2]	./src/scripts/app.ts	475 bytes	{0} [built]		
[3]	./src/scripts/quote.ts	261 bytes	{0} [built]		
[4]	./src/scripts/vendor.ts	111 bytes	{1} [built]		

```
PS C:\Student\Modules\09_NodeJS\Lab\project2> █
```

## PROJECT2

### dist

#### scripts

JS bundle.js

JS vendor.bundle.js

node\_modules

src



# Summary

- ✓ Introduction to Node.JS and Visual Studio Code
- ✓ Installing and Updating NPM packages
- ✓ Configuring Server-side Debugging Support
- ✓ Node.JS Development with TypeScript
- ✓ Using Gulp to Automate Running Tasks
- ✓ Bundling the Source Files using WebPack

