Install git & TSLint

Install version control system git, use to download/update software say from GitHub/corp server/or local installation.

1. Open browser and enter in URL <https://git-scm.com/download/win>

This should start an automatic download.

1. When download complete run downloaded file - accept all defaults
2. Check installation by opening a command window and type

git version

which should match version downloaded.

Now install TSLint, which is a used to check/enforce Typescript code quality

1. In your home directory c:\User\<corp ID> create file tslint.json and paste following into it

{

"rules": {

"class-name": true,

"comment-format": [

true,

"check-space"

],

"curly": true,

"eofline": true,

"forin": true,

"indent": [

true,

"spaces"

],

"label-position": true,

"label-undefined": true,

"max-line-length": [

true,

140

],

"member-access": false,

"member-ordering": [

true,

"static-before-instance",

"variables-before-functions"

],

"no-arg": true,

"no-bitwise": true,

"no-console": [

true,

"debug",

"info",

"time",

"timeEnd",

"trace"

],

"no-construct": true,

"no-debugger": true,

"no-duplicate-key": true,

"no-duplicate-variable": true,

"no-empty": false,

"no-eval": true,

"no-inferrable-types": true,

"no-shadowed-variable": true,

"no-string-literal": false,

"no-switch-case-fall-through": true,

"no-trailing-whitespace": true,

"no-unused-expression": true,

"no-unused-variable": true,

"no-unreachable": true,

"no-use-before-declare": true,

"no-var-keyword": true,

"object-literal-sort-keys": false,

"one-line": [

true,

"check-open-brace",

"check-catch",

"check-else",

"check-whitespace"

],

"quotemark": [

true,

"single"

],

"radix": true,

"semicolon": [

"always"

],

"triple-equals": [

true,

"allow-null-check"

],

"typedef-whitespace": [

true,

{

"call-signature": "nospace",

"index-signature": "nospace",

"parameter": "nospace",

"property-declaration": "nospace",

"variable-declaration": "nospace"

}

],

"variable-name": false,

"whitespace": [

true,

"check-branch",

"check-decl",

"check-operator",

"check-separator",

"check-type"

]

}

}

1. Open Visual Studio Code
2. Press Extensions icon, search for TSLint and press Install.
3. Open Integrated terminal and type following

npm install -g tslint typescript

1. Close Visual Studio Code (TSLint should kick in when you next open up VSC)

Download quickstart Project

This section describes how to install the barebones project quickstart from angular.io (from <https://angular.io/docs/ts/latest/quickstart.html> )

1. In a command window cd to d:\projects\vs
2. Now clone Angular 2 startup project using git:

git clone https://github.com/angular/quickstart.git quickstart

or download from <https://github.com/angular/quickstart>

1. Navigate to quickstart directory

cd quickstart

1. For now remove unnecessary files (these are for test/git…) using following commands:

for /f %i in (non-essential-files.txt) do del %i /F /S /Q

rd .git /s /q

rd e2e /s /q

1. Install all required modules

npm install

1. Launch the sample application (which uses Angular’s dev module lite-server):

npm start

You should see Hello Angular open in browser automatically

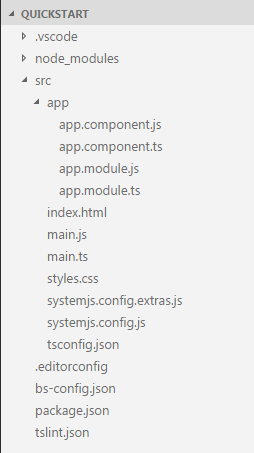
1. Stop application (Ctrl-C Y)

Configure VSC Continue

This section will describe how to configure TypeScript’s JavaScript files

1. Open folder quickstart in Visual Studio Code
2. Notice how each TypeScript file (ts) has a corresponding transpiled JavaScript file (js).

eg app.component.ts/js, app.module.ts/js, main.ts/js



1. To remove duplicated transpiled JavaScript file select File->Preferences->Settings
2. Select ‘User Settings’ and paste and save following

// Place your settings in this file to overwrite the default settings

{

"files.exclude": {

// exclude .js and .js.map files, when in a TypeScript project

"\*\*/\*.js": { "when": "$(basename).ts"},

"\*\*/\*.js.map": true

},

"files.associations": {

"\*.ejs": "html"},

"files.autoSave": "off",

"typescript.check.tscVersion": false,

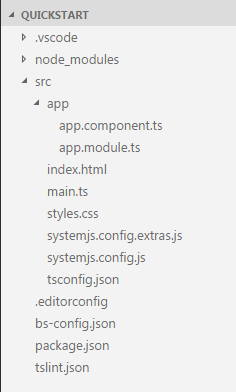
"window.zoomLevel": 0,

"[typescript]": {

}

}

1. Notice now transpiled JavaScript file (js) no longer visible from Visual Studio Code – they are still in project and get updated when their matching TypeScript (ts) file changes



Update my-app

In this section you will use

interpolation {{}}

property binding [..]

event binding (..) and

two-way binding [(ngModel)]

1. Still within quickstart folder within Visual Studio Code, open an integrated Terminal and start the Angular server in watch mode:

npm start

The welcome message should appear.

1. Update (and save) the welcome message in app.component.ts and you should see the browser message update automatically
2. To use two-way model binding need to add module FormsModel (it is not part of the default installation). Update app.module.ts to



1. Now update app.component.ts – note the use of the ` back-tick (ES2005)



1. Check

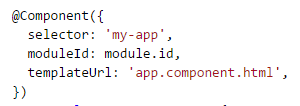
Event & property bindings working - show/hide

Interpolation & two-way binding working - ‘message’ changes

1. Html is getting rather large so create a file app.component.html and cut and paste everything from the two back-ticks in app. component.ts to new file app.component.html (ensure it is created at same level as app.component.ts).
2. Modify app.component.ts to reference app.component.html

templateUrl: 'app/app.component.html',

1. Test app still working.
2. Rather than use the full path to html file, add module.id which appends the path to html file: Change to:



1. Test app still working