TD 3 – Angular-Cli + Web IDE + Chrome DevTools

TD Objectives

During this hands-on, you will get familiar to basic Web Angular development, use Chrome Developer Tools, and use a rich web IDE : WebStorm (you may find equivalent features in Visual Studio Code, or Eclipse)

Step 1 : setup new Angular project, Import in WebStorm IDE

Pre-requisites: check you have nodejs installed, and angular cli (npm install -g @angular/cli)

Setup a new Angular project, using

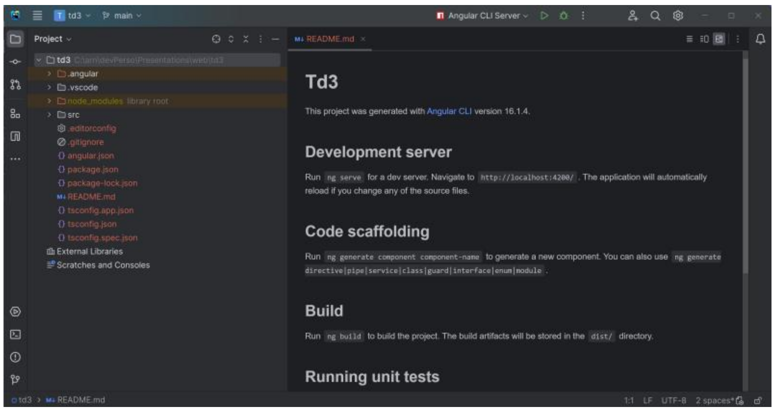
```
C:\td3> ng new
```

When prompted to use Angular rooting … use "Y"   (override default "N" !)

When prompted for stylesheet format, use CSS.

This may take 5mn depending on your network bandwidth.

Then open your project from WebStorm IDE.



Td3

This project was generated with Angular CLI version 16.1.4.

Development server

Run `ng serve` for a dev server. Navigate to `http://localhost:4200/`. The application will automatically reload if you change any of the source files.

Code scaffolding

Run `ng generate component component-name` to generate a new component. You can also use `ng generate directive|pipe|service|class|guard|interface|enum|module`.

Build

Run `ng build` to build the project. The build artifacts will be stored in the `dist/` directory.

Running unit tests

[Optional] Step 1 (next) : setup git

It is highly recommended to setup git for versioning your work.

By committing often your files, you have backup, and undo/redo history.

You can also save it on http://github.com or gitlab

By using "ng new", you already have ".gitignore" file correctly configured (containing line "/node_modules"), so to finalize you git project, use only need to type

```
C:\td3> git init

C:\td3> git add .

C:\td3> git commit -m "init td3"
```

Then regularly use these git commands (maybe from your IDE, or github desktop app)

git add .

git commit -m "some message"

git diff

git log

git push

etc …

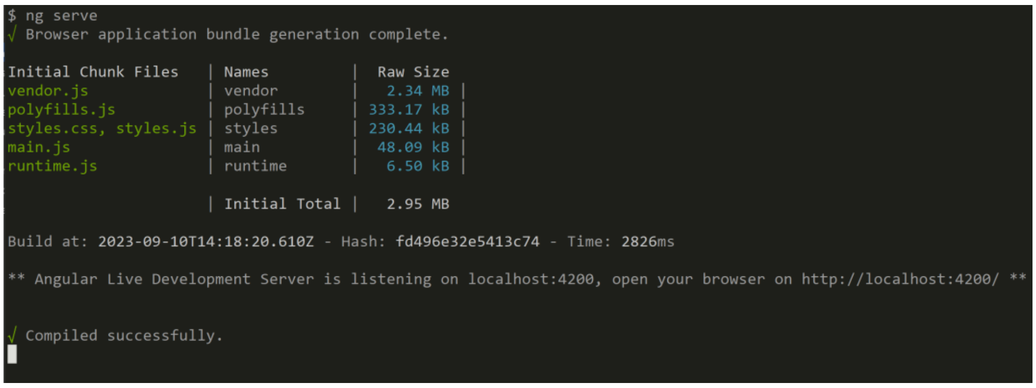
Step 2 : launch ng serve, Debug in Chrome DevTools

Launch your ng serve, using

C:\td3> ng serve

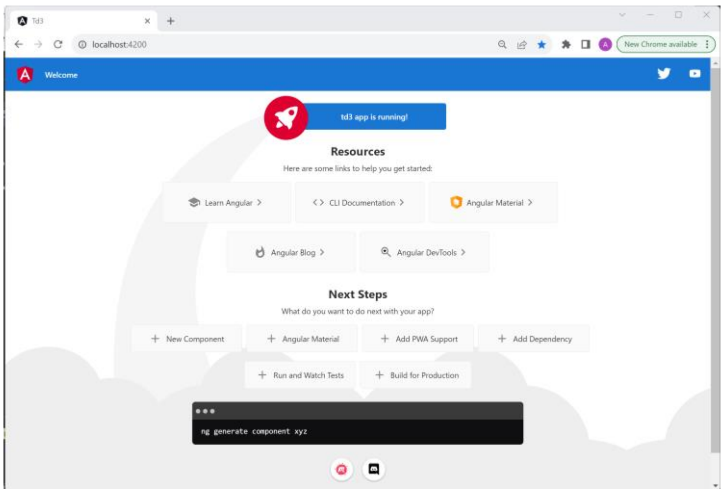
(or alternatively "npm run start" or "npm run watch")

At this point, you should see



```
$ ng serve
√ Browser application bundle generation complete.

Initial Chunk Files   | Names         |  Raw Size
vendor.js             | vendor        |   2.34 MB |
polyfills.js          | polyfills     | 333.17 kB |
styles.css, styles.js | styles        | 230.44 kB |
main.js               | main          |  48.09 kB |
runtime.js            | runtime       |   6.50 kB |

                      | Initial Total |   2.95 MB

Build at: 2023-09-10T14:18:20.610Z - Hash: fd496e32e5413c74 - Time: 2826ms

** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **

√ Compiled successfully.
```

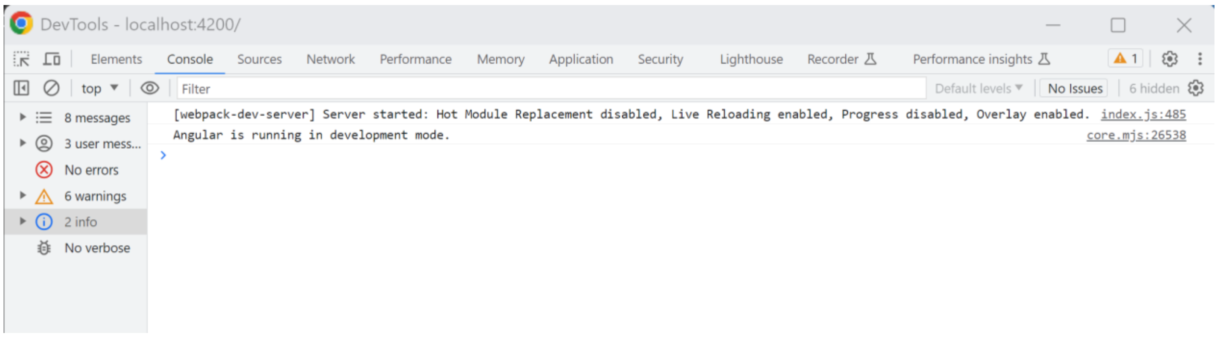
Once started, open you web browser(Chrome) on http://localhost:4200



Open Chrome Developer Tools (F12 or Ctrl+Shitf+I)

Step 2- A/ use DevTools > console tab

Ensure you are able to see logs in the console tab (default tab to be opened)

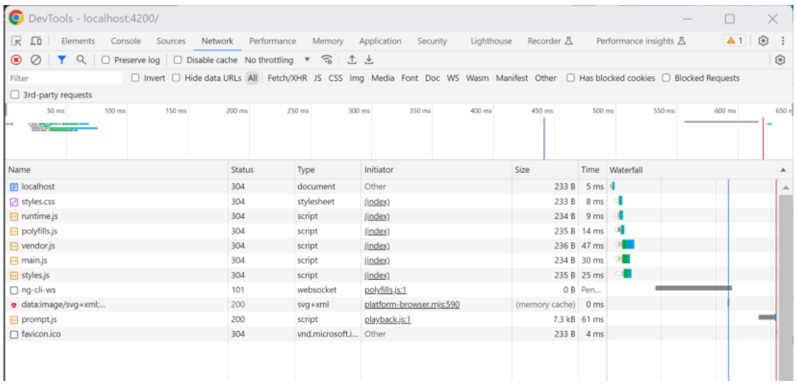


```
[webpack-dev-server] Server started: Hot Module Replacement disabled, Live Reloading enabled, Progress disabled, Overlay enabled.  index.js:485
Angular is running in development mode.                                                                                       core.mjs:26538
```

We will see in next questions how to add more logs in your code.

Step 2 – B / use DevTools > Network tab

refresh page and see "http GET" for /index.html

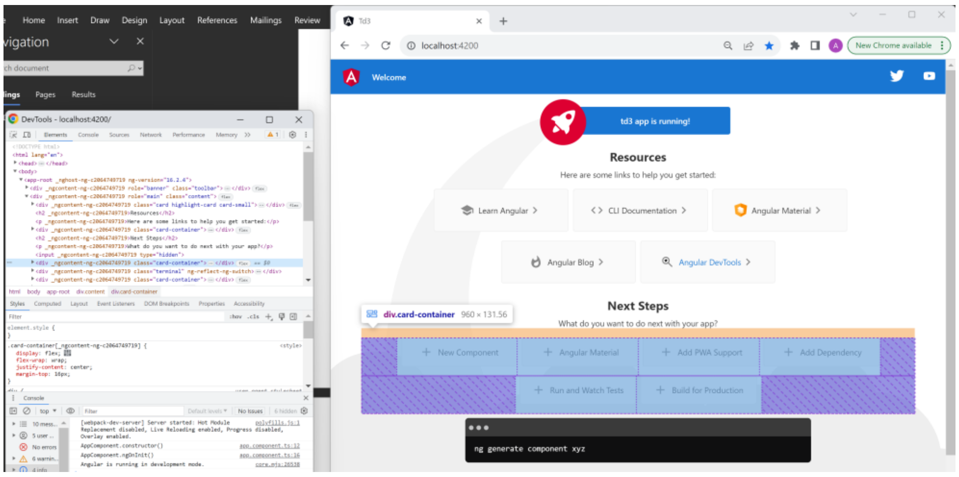


Name	Status	Type	Initiator	Size	Time
localhost	304	document	Other	233 B	5 ms
styles.css	304	stylesheet	(index)	233 B	8 ms
runtime.js	304	script	(index)	234 B	9 ms
polyfills.js	304	script	(index)	235 B	14 ms
vendor.js	304	script	(index)	236 B	47 ms
main.js	304	script	(index)	234 B	30 ms
styles.js	304	script	(index)	235 B	25 ms
ng-cli-ws	101	websocket	polyfills.js:1	0 B	Pen...
data:image/svg+xml;...	200	svg+xml	platform-browser.mjs:590	(memory cache)	0 ms
prompt.js	200	script	playback.js:1	7.3 kB	61 ms
favicon.ico	304	vnd.microsoft.i...	Other	233 B	4 ms

This tab is specially important when debugging interaction between your front-end (angular) and your backend Rest api server, but not in this TD (see next TD).

Step 2 – C/ use DevTools > Element layout tab

Try to select the div panel below the "Next Steps" section



```
<!DOCTYPE html>
<html lang="en">
 ▸<head> … </head>
 ▾<body>
   ▾<app-root _nghost-ng-c2064749719 ng-version="16.2.4">
     ▸<div _ngcontent-ng-c2064749719 role="banner" class="toolbar"> … </div> flex
     ▾<div _ngcontent-ng-c2064749719 role="main" class="content"> flex
       ▸<div _ngcontent-ng-c2064749719 class="card highlight-card card-small"> … </div> flex
        <h2 _ngcontent-ng-c2064749719>Resources</h2>
        <p _ngcontent-ng-c2064749719>Here are some links to help you get started:</p>
       ▸<div _ngcontent-ng-c2064749719 class="card-container"> … </div> flex
        <h2 _ngcontent-ng-c2064749719>Next Steps</h2>
        <p _ngcontent-ng-c2064749719>What do you want to do next with your app?</p>
        <input _ngcontent-ng-c2064749719 type="hidden">
       ▸<div _ngcontent-ng-c2064749719 class="card-container"> … </div> flex == $0
       ▸<div _ngcontent-ng-c2064749719 class="terminal" ng-reflect-ng-switch> … </div>
       ▸<div _ngcontent-ng-c2064749719 class="card-container"> … </div> flex
```

```
element.style {
}
.card-container[_ngcontent-ng-c2064749719] {                <style>
  display: flex;
  flex-wrap: wrap;
  justify-content: center;
  margin-top: 16px;
}
```

```
[webpack-dev-server] Server started: Hot Module          polyfills.js:1
Replacement disabled, Live Reloading enabled, Progress disabled,
Overlay enabled.
AppComponent.constructor()                               app.component.ts:12
AppComponent.ngOnInit()                                  app.component.ts:16
Angular is running in development mode.                  core.mjs:26538
```

```
ng generate component xyz
```

When selecting a Div either from mouse or selecting in the html, you get the mapping between html source code and how elements are displayed. You can also debug the CSS (border, margin, padding size..), and even change them directly in Chrome. Try it.

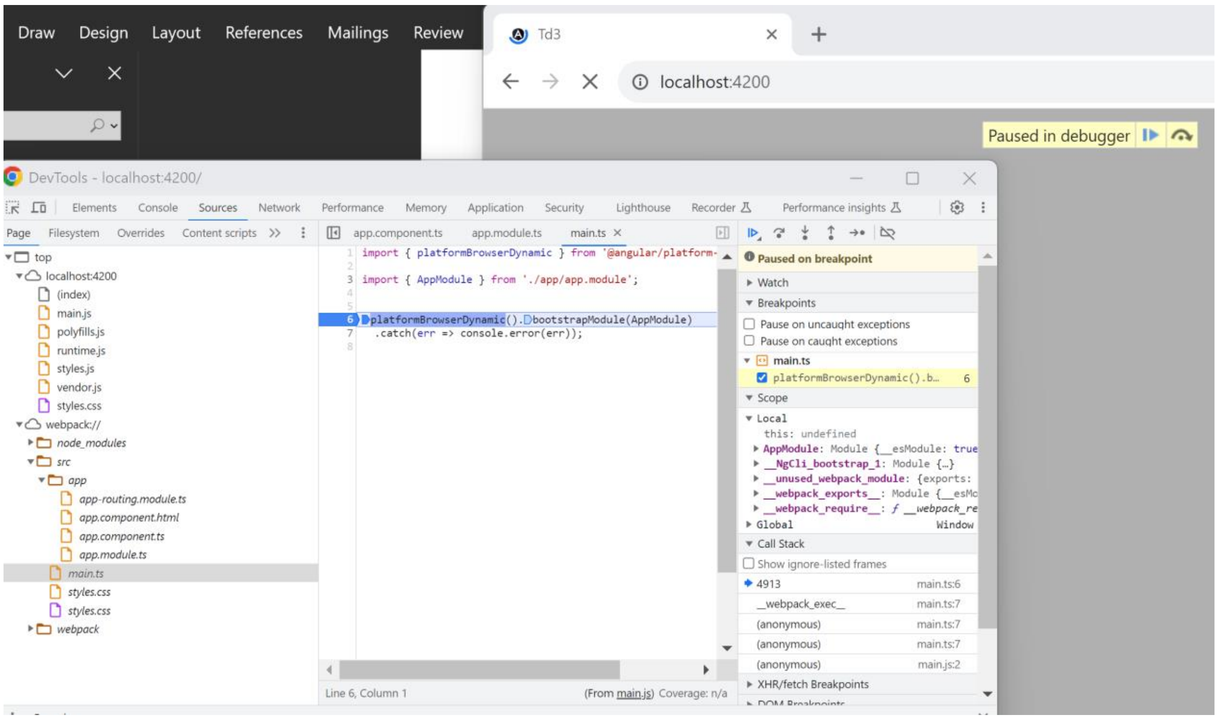
Step 2 – D / use DevTools > Source tab

From Chrome > Source tab, open the left element "webpack://", then sub-element "src", "app", "main.ts"

Put a breakpoint in line 6 by clicking on the left margin on the main.ts file in the editor.

When refreshing your web page… This is the very first line that Angular execute.

Your breakpoint should cause your page to be "Paused in debugger"



```
1  import { platformBrowserDynamic } from '@angular/platform-
2
3  import { AppModule } from './app/app.module';
4
5
6  platformBrowserDynamic().bootstrapModule(AppModule)
7    .catch(err => console.error(err));
8
```

In reality, you program is pausing in javascript file "main.js" (under top > localhost:4200 >main.js), but fortunately in development mode, it is correctly configured to use "source map" file, so you browse in "top > webpack:// " !

Step 3 : modify app.ts … use your IDE autocompletion features

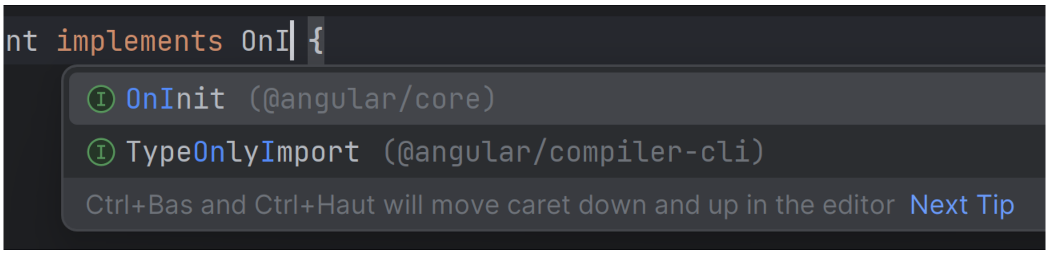
Go back in your WebStorm IDE, edit your file "src/app/app.component.ts"

Add a "constructor() { }"   method to the class (it is implicit when none is found). Put "console.log('AppComponent.constructor()')" in the constructor body.

Check you are now able to see the console message in Chrome DevTool

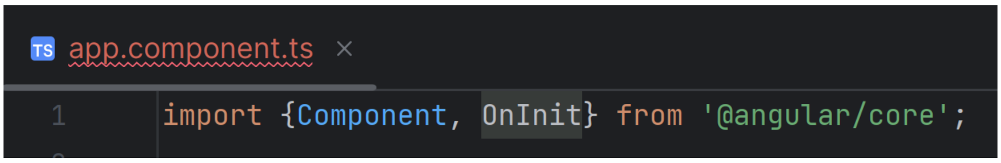
You may also add console.log in method ngOnInit() {..} and add import + implements OnInit interface.

The expected code is below, but do not copy&paste … try to use auto-completion features from your IDE



```
nt implements OnI {
  OnInit (@angular/core)
  TypeOnlyImport (@angular/compiler-cli)
  Ctrl+Bas and Ctrl+Haut will move caret down and up in the editor  Next Tip
```

From your IDE, when selecting interface OnInit, it should automatically add corresponding import !

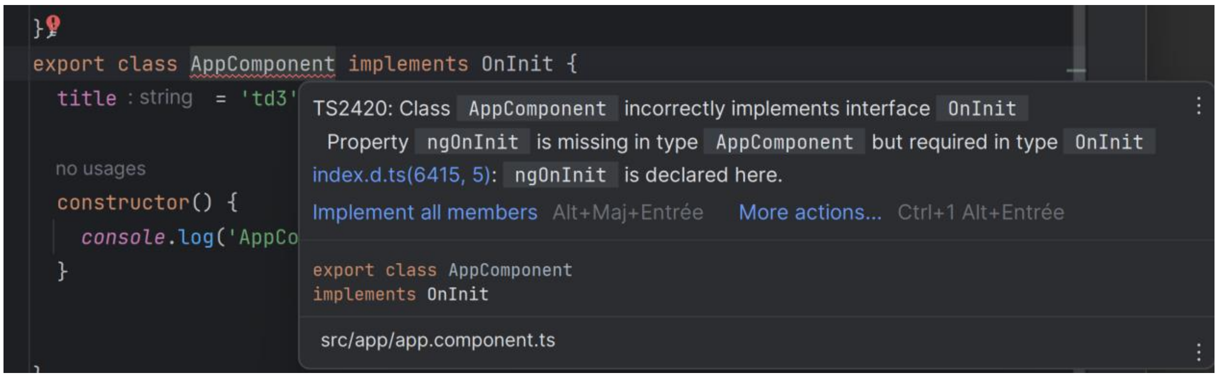


app.component.ts

```
import {Component, OnInit} from '@angular/core';
```

Finally, implement the requested method … using automatic error fixing

Either click on underlined error on "AppComponent", or click on red light



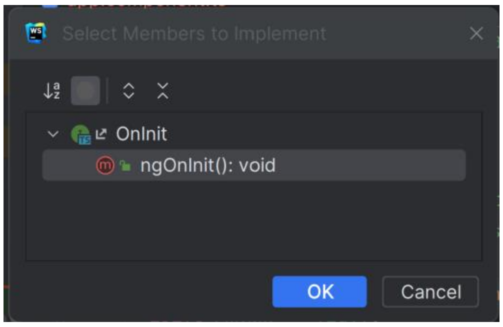
```
}
export class AppComponent implements OnInit {
  title : string = 'td3'
  no usages
  constructor() {
    console.log('AppCo
  }
```

TS2420: Class AppComponent incorrectly implements interface OnInit
Property ngOnInit is missing in type AppComponent but required in type OnInit
index.d.ts(6415, 5): ngOnInit is declared here.
Implement all members Alt+Maj+Entrée   More actions... Ctrl+1 Alt+Entrée

```
export class AppComponent
implements OnInit
```

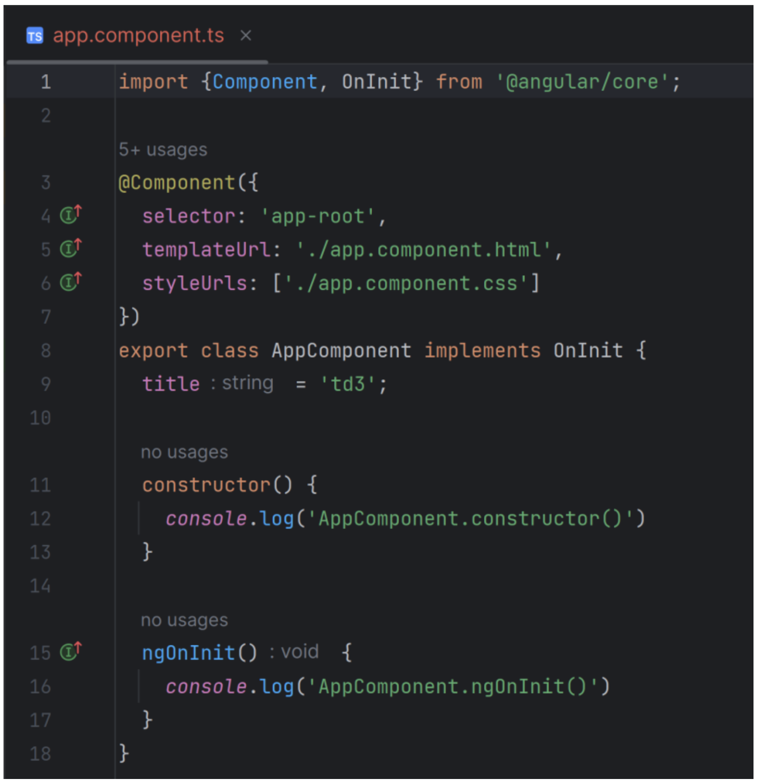
src/app/app.component.ts

Select all missing method to implement:



Finish editing ngOnInit() {.. }, add another "console.log('AppComponent.ngOnInit()");"

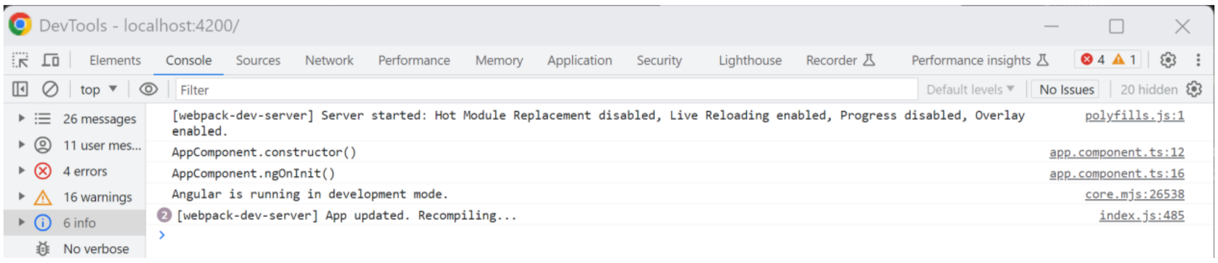
You should have:



```typescript
import {Component, OnInit} from '@angular/core';

5+ usages
@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent implements OnInit {
  title : string  = 'td3';

  no usages
  constructor() {
    console.log('AppComponent.constructor()')
  }

  no usages
  ngOnInit() : void  {
    console.log('AppComponent.ngOnInit()')
  }
}
```

Now re-check you logs in Chrome DevTools > console.

You should see both lines :



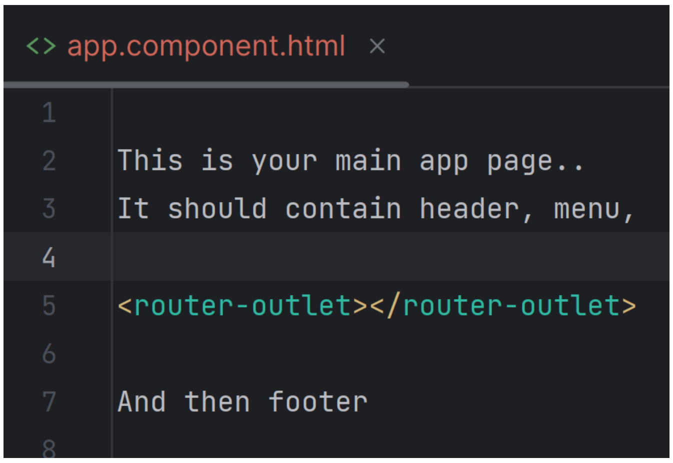
```
[webpack-dev-server] Server started: Hot Module Replacement disabled, Live Reloading enabled, Progress disabled, Overlay    polyfills.js:1
enabled.
AppComponent.constructor()                                                                                         app.component.ts:12
AppComponent.ngOnInit()                                                                                            app.component.ts:16
Angular is running in development mode.                                                                               core.mjs:26538
2 [webpack-dev-server] App updated. Recompiling...                                                                      index.js:485
>
```

Remark :

- Constructor code in Angular classes should not perform any angular operation … constructor is only for allocating object memory in Javascript. This is why we rather use “ngOnInit”, which is called (after constructor), when the object is ready to be used
- When the object is about to be destroyed, there is a symmetric method… ngOnDestroy, in interface OnDestroy. Try implementing it (but it won’t be used yet because your app is not changing page route yet)

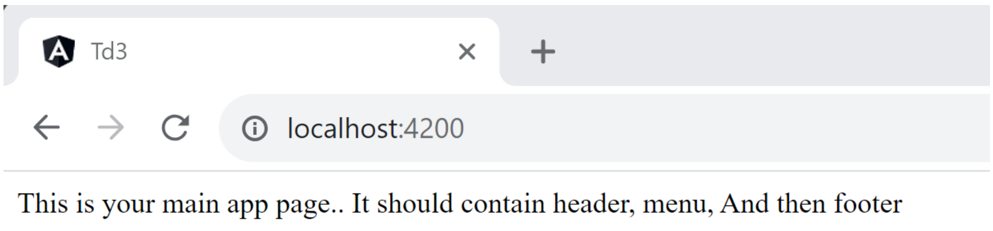
Step 4: Modify main app.component.html component page

Modify your app.component.html page like below



```
<> app.component.html ×

This is your main app page..
It should contain header, menu,

<router-outlet></router-outlet>

And then footer
```

From Chrome, you should see this:



This is your main app page.. It should contain header, menu, And then footer

Remark 1: the page in Chrome changes in real-time, while saving. (this is "auto-reload")

Remark 2: Notice you should not remove the <router-outlet></router-outlet> element, it will be used later

Step 5: Add ng-bootstrap dependency

Interrupt your running "ng serve" (Control+C)

Add "ng-bootstrap" as a npm dependency only… This will not work (on purpose), we will see right after why

```
C:\td3>  npm install –save @ng-bootstrap/ng-bootstrap
```

Check what are the source files that have been modified by this operation, using "git diff"

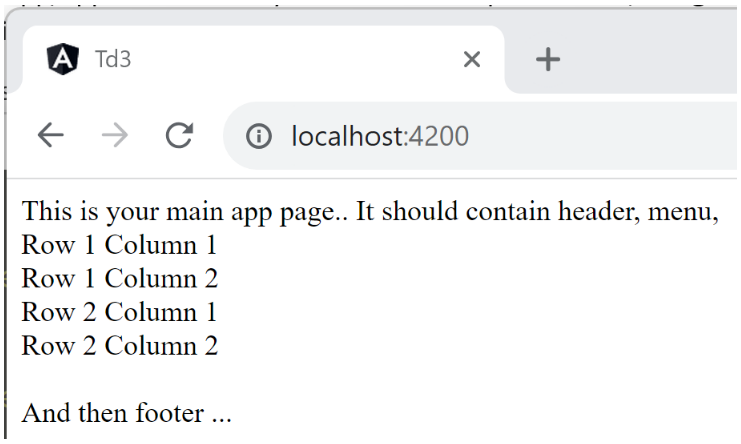
Then restart again your "ng serve"

Edit file "src/app/app.html" to test your first bootstrap html code, using nested divs with class="container", then class="row", class="col-md-2" (or col-md-3, col-md-4, col-md-6..) .

Your html result should (but does not yet…) look like this:

Row1 colum 1	row 1 column 2
Row2 colum 1	row 2 column2

Unfortunately, the layout does not work (as expected) … all elements are above each other, not forming a layout with rows and columns



This is your main app page.. It should contain header, menu,
Row 1 Column 1
Row 1 Column 2
Row 2 Column 1
Row 2 Column 2

And then footer ...

Redo the correct installation of "ng add", instead of "npm install", using

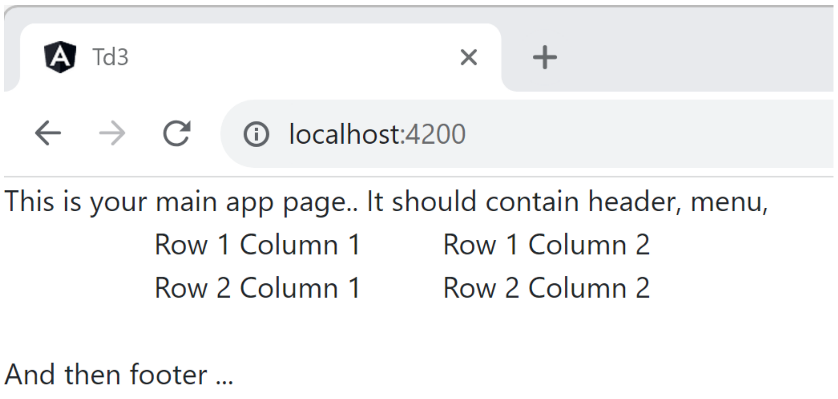
C:\td3> ng add @ng-bootstrap/ng-bootstrap

Check again what are the files that have been modified in git

See in file angular.json this added line ?

```
"node_modules/bootstrap/dist/css/bootstrap.min.css",
```

Now restart "ng serve", and refresh your page, you should see this:



This is your main app page.. It should contain header, menu,
Row 1 Column 1 Row 1 Column 2
Row 2 Column 1 Row 2 Column 2

And then footer ...

Redo the same (copy&paste below) using class="container-fluid" instead of class="container"

Find out the difference between class="container" and class="container-fluid". Are they using same CSS width on the page ?

Step 6: change basic CSS, to add debugging border style

Edit your file src/style.css

.app-border {

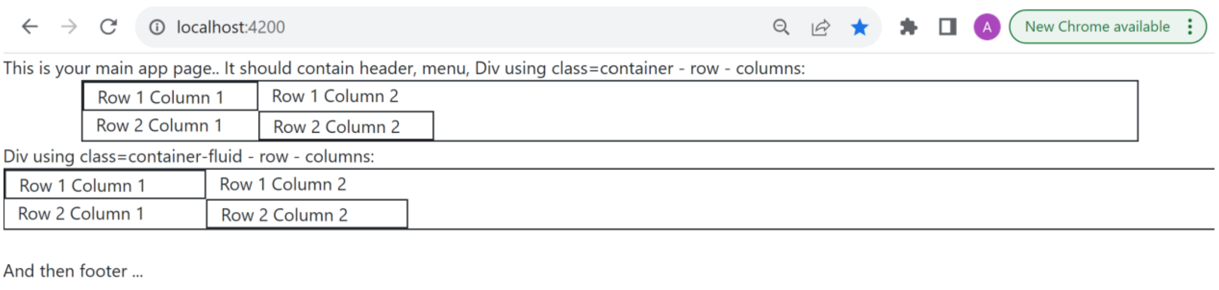
border-style: solid;

border-width: 1pt;

}

Then add this extra class "app-border" to the top level div with class="container" and also to the divs of "row1 column1" and to the div of "row 2 - column 2"

At this point, you should get:


localhost:4200

This is your main app page.. It should contain header, menu, Div using class=container - row - columns:

Row 1 Column 1 | Row 1 Column 2
Row 2 Column 1 | Row 2 Column 2

Div using class=container-fluid - row - columns:

Row 1 Column 1 | Row 1 Column 2
Row 2 Column 1 | Row 2 Column 2

And then footer ...

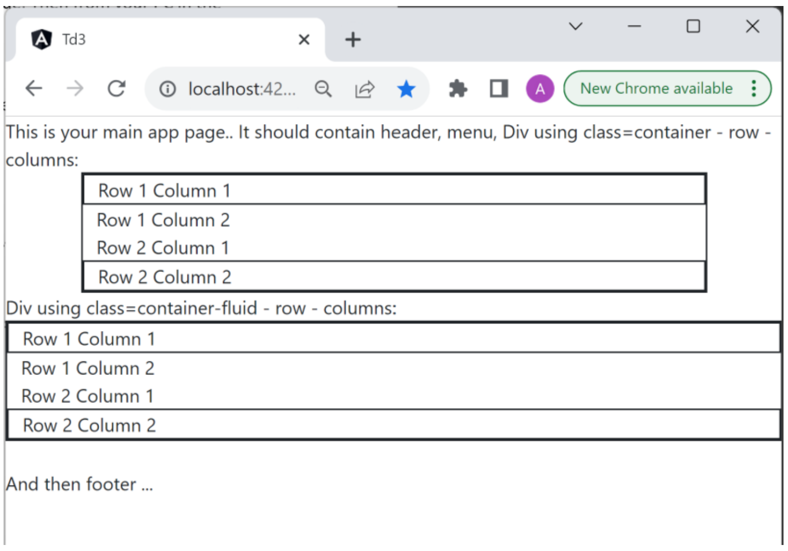
With these debugging "app-border" class, you can easily see you div size by forcing a temporary border to it.

Step 7: Resize your Chrome window … see what happen on Small/Medium/Large size

When using your web page from your Portable telephone, you are in the "Small" device mode.

When using from your Tablet, you are in the "Medium" device mode. Then from your PC in the "Large" device mode.

When resizing to a small window, bootstrap does not any more use columns, but display all cells vertically. You should see this:


Td3

localhost:42...

This is your main app page.. It should contain header, menu, Div using class=container - row - columns:

Row 1 Column 1
Row 1 Column 2
Row 2 Column 1
Row 2 Column 2

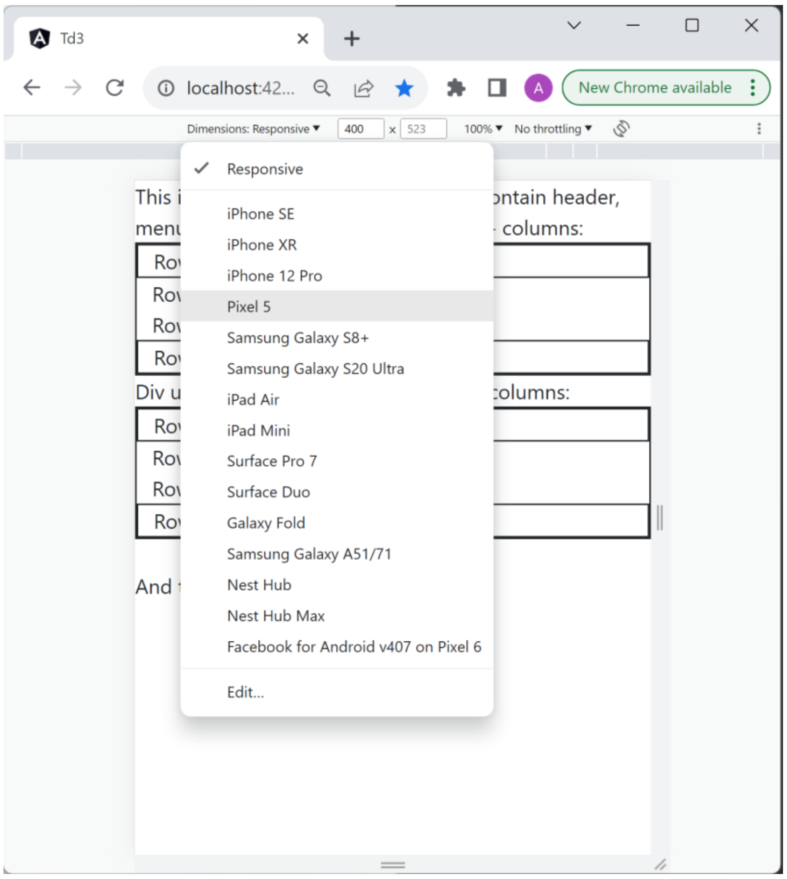
Div using class=container-fluid - row - columns:

Row 1 Column 1
Row 1 Column 2
Row 2 Column 1
Row 2 Column 2

And then footer ...

Play with alternate bootstrap classes for small / medium / large devices.

Chrome also has tool for debugging custom dimensions for well-known devices.



Dimensions: Responsive ▾ 400 x 523 100% ▾ No throttling ▾

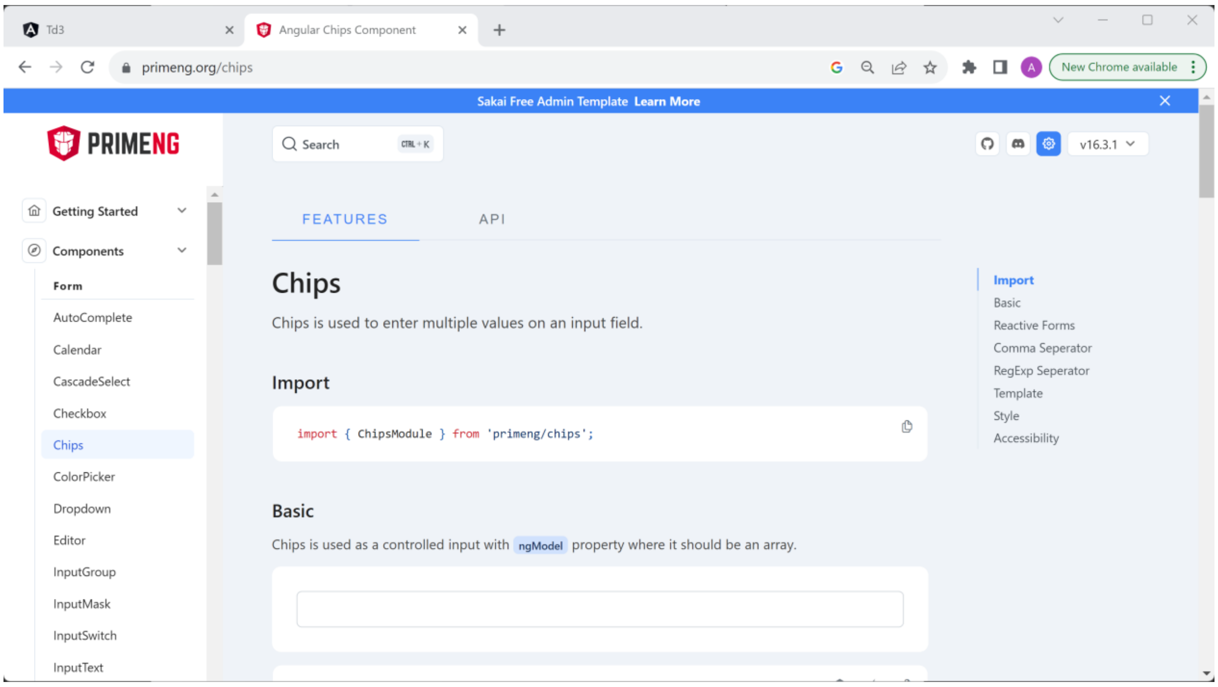
Step 8: Add other dependency : primeNg … play with “chips” component

Add in your project the dependency correcponding to the “Chips” component of the “primeNG” library

See the doc and online component ready to use:

https://primeng.org/chips

Chips is a rich text area that support entering several tags easily (like “badges” in bootstrap)



Chips

Chips is used to enter multiple values on an input field.

Import

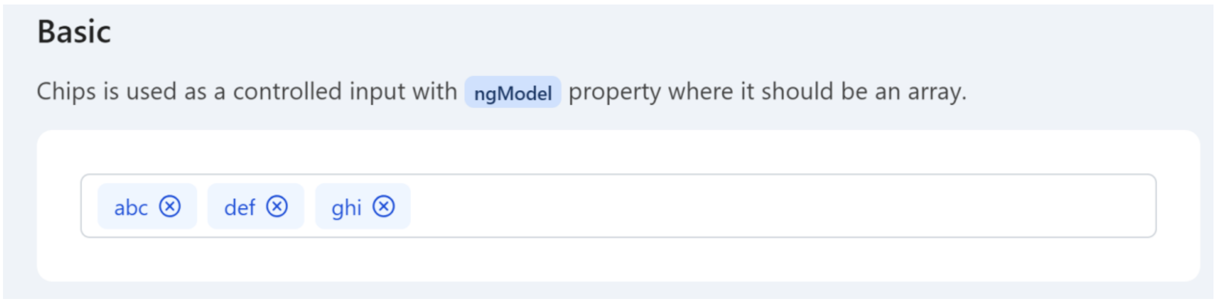
```
import { ChipsModule } from 'primeng/chips';
```

Basic

Chips is used as a controlled input with ngModel property where it should be an array.

Test yourself the online “Chips” component by typing “abc <enter> def <enter> ghi <enter>”

You should get this:



Basic

Chips is used as a controlled input with ngModel property where it should be an array.

Now edit your project to have this component directly in you app.html page

Follow the steps described in the doc:

1/ add npm dependency … restart ng serve

npm install -s primeng

2/ edit your angular.json, add lines

```
"styles": [
   "node_modules/primeng/resources/themes/lara-light-blue/theme.css",
   "node_modules/primeng/resources/primeng.min.css",
   ...
]
```

3/ edit your app/app.module.ts to add

```
import { FormsModule } from '@angular/forms';
import { ChipsModule } from 'primeng/chips';

@NgModule({
  imports: [
        ...
        FormsModule,
        ChipsModule
  ],
```

4/ edit your app.component.html, to add

```
<p-chips [(ngModel)]="values"></p-chips>
```

5/ edit your app.component.ts, add

```
values: string ='';
```

Check that it works

Step 9: Browse PrimeNG source code

From WebStorm, click on '<p-prime>' html element, and navigate into declaration

You see all the code interface definition (file "node_modules/primeng/chips/chips.d.ts")

Unfortunatly, you do not see real source code!

Check that the file use

```
export declare class Chips { ... }
```

instead of

```
export class Chips { ... }
```

To see real source code, as it is Open-Source, you can still browse to

https://github.com/primefaces/primeng/blob/master/src/app/components/chips/chips.ts#L73

Read the code to get an idea of how complex it can be to write a rich component, supporting keyboard handling, drag&drop, copy&paste, etc..