

Angular2: import external js file into component

Asked 3 years, 3 months ago Active 10 days ago Viewed 132k times

I'm going to import this [d3gauge.js](#) file into one of my angular2 component, `memmon.component.ts` file.

```
55 import '../../../js/d3gauge.js';
   export class MemMonComponent {
       createMemGauge() {
           new drawGauge(this.opt); //drawGauge() is a function inside d3gauge.js
       }
   }
```

and in the corresponding template file, add

```
<script src="../../../js/d3gauge.js"></script>
```

But it doesn't work, `drawGauge` can't be found.

So,

1. what're correct steps to import an external js file to angular2?
2. since I'm using webpack, is it possible to do it in webpack? I refer to this [question](#) , the webpack solution there doesn't work for me as a result of `.ensure` can't be resolved.

javascript angular webpack

asked May 6 '16 at 21:52



Bing Lu

1,134 5 21 29

2 you need to create or import typings (*.d.ts) for the file. – [ps2goat](#) May 6 '16 at 22:45

8 Answers

Ideally you need to have `.d.ts` file for typings to let `Linting` work.

65

But It seems that `d3gauge` doesn't have one, you can Ask the developers to provide and hope they will listen.



Alternatively, you can solve this specific issue by doing this

```
declare var drawGauge: any;

import '../.../js/d3gauge.js';
export class MemMonComponent {
  createMemGauge() {
    new drawGauge(this.opt); //drawGauge() is a function inside d3gauge.js
  }
}
```

If you use it in multiple files, you can create a `d3gauge.d.ts` file with the content below

```
declare var drawGauge: any;
```

and reference it in your `boot.ts` (bootstrap) file at the top, like this

```
///

```

answered May 7 '16 at 4:39



Ankit Singh

17.9k 4 50 80

- 1 Thanks. One more question, when will the `declare var drawGauge: any;` be referenced, at the moment of transpile this ts file? Or, put it in another way, when `new drawGauge(this.opt)`, how does the compiler know the `drawGauge()` is actually in `d3gauge.js`? – Bing Lu May 7 '16 at 4:56

Yes, on transpile, it has to know that there is/ or will be something named `d3gauge`. In second question, it doesn't, `d3gauge.js` assigns it to a `global` variable in the `dom` and that is from where it's being called on runtime. – Ankit Singh May 7 '16 at 5:05

- 1 just put `<script src="../../.../js/d3gauge.js"></script>` in your `index.html` only, nowhere else, webpack solution probably doesn't work because of wrong path, but i don't know anything about webpack – Ankit Singh May 7 '16 at 16:00

- 1 @BingLu Hey man, did you get the solution working for you? I still got the `EXCEPTION: Uncaught (in promise): Error: Error in someComponent saying that the declared variable is not defined and could not find a solution so far.` – Mr.Moe Sep 26 '16 at 10:18

- 1 An answer worth an upvote... Thanks, buddy...! – Mr. Noddy Jun 20 '18 at 7:24



42



After wasting a lot of time in finding its solution, **I've found one**. For your convenience I've used the complete code that you can replace your whole file with.

This is a general answer. Let's say you want to import a file named testjs.js into your angular 2 component. Create testjs.js in your assets folder:

assets > testjs.js

```
function test(){  
    alert('TestingFunction')  
}
```

include testjs.js in your index.html

index.html

```
<!doctype html>  
<html>  
<head>  
  <meta charset="utf-8">  
  <title>Project1</title>  
  <base href="/">  
  <meta name="viewport" content="width=device-width, initial-scale=1">  
  <link rel="icon" type="image/x-icon" href="favicon.ico">  
  
  <script src="./assets/testjs.js"></script>  
  
</head>  
<body>  
  <app-root>Loading...</app-root>  
</body>  
</html>
```

In your app.component.ts or in any component.ts file where you want to call this js declare a variable and call the function like below:

app.component.ts

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

```
declare var test: any;
```

```
@Component({  
  selector: 'app-root',  
  templateUrl: './app.component.html',  
  styleUrls: ['./app.component.css']  
})
```

```
export class AppComponent {  
  title = 'app works!';
```

```
  f(){  
    new test();  
  }  
}
```

Finally in your **app.component.html** test the function

app.component.html

```
<h1>
  <button (click)='f()'>Test</button>
</h1>
```

answered Jun 18 '17 at 15:48



Muhammad Rehan Qadri

1,178 11 14

- 4 this is the answer I am looking for. Thanks a ton, can you just share why we need 'new' here ? – [habibalsaki](#) Jul 29 '17 at 18:47

do we need to declare the function name on component, what if i have many functions do I need to declare all? Also its not working i am getting these errors "ERROR ReferenceError: test is not defined" and "Refused to execute script from 'localhost:4200/assets/custom.js' because its MIME type ('text/html') is not executable, and strict MIME type checking is enabled." – [Sunil Chaudhary](#) May 25 '18 at 5:37

Is it necessary that my javascript files should be in asset folder? Can't it be in any other folder? – [Akhil Nair](#) Jan 25 at 12:53

Instead of including your `js` file extension in `index.html` , you can include it in `.angular-cli.json` file.

9

These are the steps I followed to get this working:

1. First include your external `js` file in `assets/js`
2. In `.angular-cli.json` - add the file path under scripts: `[../app/assets/js/test.js]`
3. In the component where you want to use the functions of the `js` file.

Declare at the top where you want to import the files as

```
declare const Test:any;
```

After this you can access its functions as for example `Test.add()`

edited Aug 8 at 21:34



llaaalu

142 2 10

answered Feb 9 '18 at 12:22



Sitaram

139 2 4

Hi, can you tell me where to put the part "Declare const Test:any;" ? Inside the component? I tried to put it inside the export class, and also before the "export class...." but it didn't work. – [Vladimir Despotovic](#) Aug 7 '18 at 13:21

In your `app.component.ts` or in any `component.ts` file where you want to call the method of Test, just add "Declare const Test:any" at the top where you import other dependencies of that component. – [Sitaram](#) Aug 9 '18 at 7:05

declare not Declare – [Jnr](#) May 15 at 17:17

The following approach worked in Angular 5 CLI.

5 For sake of simplicity, I used similar d3gauge.js demo created and provided by oliverbinns - which you may easily find on Github.

So first, I simply created a new folder named **externalJS** on same level as the **assets** folder. I then copied the 2 following .js files.

- d3.v3.min.js
- d3gauge.js

I then made sure to declare both linked directives in main **index.html**

```
<script src="./externalJS/d3.v3.min.js"></script>
<script src="./externalJS/d3gauge.js"></script>
```

I then added a similar code in a **gauge.component.ts** component as followed:

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

declare var d3gauge:any; <----- !
declare var drawGauge: any; <-----!

@Component({
  selector: 'app-gauge',
  templateUrl: './gauge.component.html'
})

export class GaugeComponent implements OnInit {
  constructor() { }

  ngOnInit() {
    this.createD3Gauge();
  }

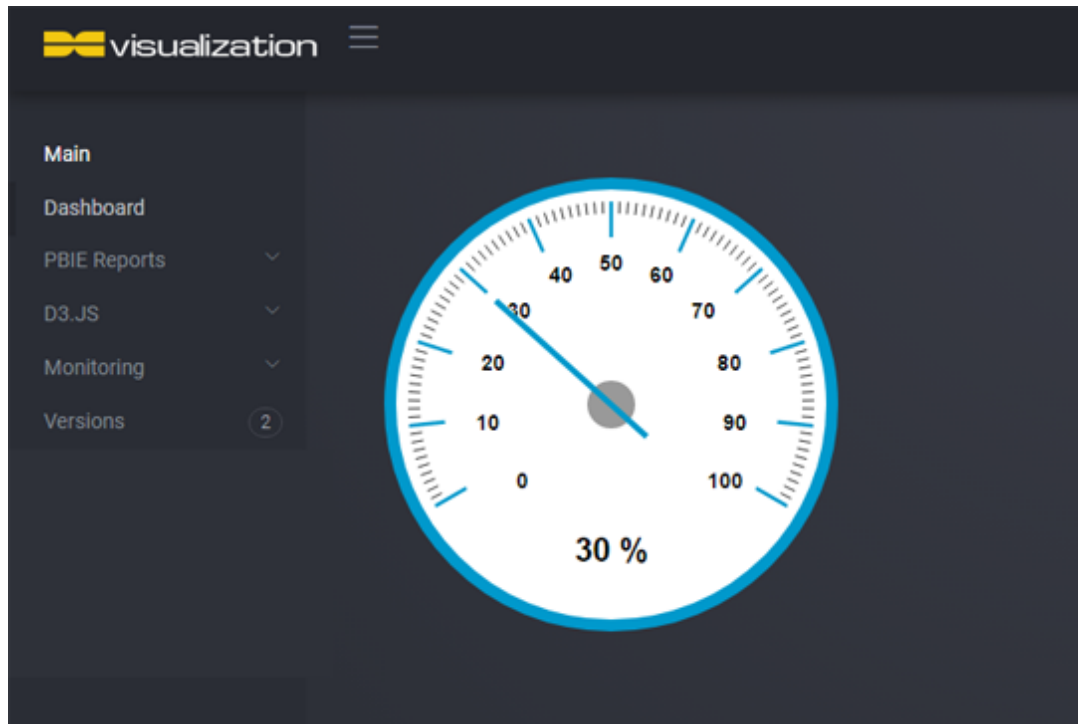
  createD3Gauge() {
    let gauges = []
    document.addEventListener("DOMContentLoaded", function (event) {
      let opt = {
        gaugeRadius: 160,
        minVal: 0,
        maxVal: 100,
        needleVal: Math.round(30),
        tickSpaceMinVal: 1,
        tickSpaceMajVal: 10,
        divID: "gaugeBox",
        gaugeUnits: "%"
      }

      gauges[0] = new drawGauge(opt);
    });
  }
}
```

and finally, I simply added a div in corresponding gauge.component.html

```
<div id="gaugeBox"></div>
```

et voilà ! :)



answered Dec 18 '17 at 18:05



PeteZaria

149 2 7

Here is a simple way i did it in my project.

2

lets say you need to use `clipboard.min.js` and for the sake of the example lets say that inside `clipboard.min.js` there is a function that called `test2()` .

in order to use `test2()` function you need:

1. make a reference to the .js file inside you index.html.
2. import `clipboard.min.js` to your component.
3. declare a variable that will use you to call the function.

here are only the relevant parts from my project (see the comments):

index.html:

```
<!DOCTYPE html>
<html>
<head>
  <title>Angular QuickStart</title>
  <base href="/src/">
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="styles.css">

  <!-- Polyfill(s) for older browsers -->
```

```

<script src="/node_modules/core-js/client/shim.min.js"></script>

<script src="/node_modules/zone.js/dist/zone.js"></script>
<script src="/node_modules/systemjs/dist/system.src.js"></script>

<script src="systemjs.config.js"></script>
<script>
  System.import('main.js').catch(function (err) { console.error(err); });
</script>

<!-- ***** HERE IS THE REFERENCE TO clipboard.min.js -->
<script src="app/txtzone/clipboard.min.js"></script>
</head>

<body>
  <my-app>Loading AppComponent content here ...</my-app>
</body>
</html>

```

app.component.ts:

```

import '../txtzone/clipboard.min.js';
declare var test2: any; // variable as the name of the function inside clipboard.min.js

@Component({
  selector: 'txt-zone',
  templateUrl: 'app/txtzone/Txtzone.component.html',
  styleUrls: ['app/txtzone/TxtZone.css'],
})

export class TxtZoneComponent implements AfterViewInit {

  // call test2
  callTest2()
  {
    new test2(); // the javascript function will execute
  }

}

```

answered Apr 1 '18 at 13:53



jonathana

3,940 3 13 29

You can also try this:

1 `import * as drawGauge from '.././.././././js/d3gauge.js';`

and just `new drawGauge(this.opt);` in your ts-code. This solution works in project with angular-cli embedded into laravel on which I currently working on. In my case I try to import `poliglot` library (btw: very good for translations) from `node_modules`:

```
import * as Polyglot from '.././../node_modules/node-polyglot/build/polyglot.min.js';
...
export class Lang
{
  constructor() {
    this.polyglot = new Polyglot({ locale: 'en' });
    ...
  }
  ...
}
```

This solution is good because i **don't need to COPY** any files from `node_modules` :).

UPDATE

You can also look on [this LIST](#) of ways how to include libs in angular.

edited Jul 8 '18 at 17:12

answered May 24 '18 at 18:31



Kamil Kielczewski

19.5k 8 91 113

Let's say you have added a file "xyz.js" under *assets/js* folder in some *Angular* project in *Visual-Studio*, then the easiest way to include that file is to add it to **.angular-cli.json**

1

```
"scripts": [ "assets/js/xyz.js" ],
```

You should be able to use this JS file's functionality in your component or .ts files.

edited Jun 20 at 15:59

answered Jun 20 at 15:46



Stphane

2,860 1 23 34



Anuj Vohra

11 2

1) First Insert JS file path in an **index.html** file :

1

```
<script src="assets/video.js" type="text/javascript"></script>
```

2) Import JS file and declare the variable in **component.ts** :

- import `'././././assets/video.js'`;
- declare var **RunPlayer**: any;

NOTE: Variable name should be same as the name of a function in js file

3) Call the js method in the component

```
ngAfterViewInit(){
  setTimeout(() => {
    new RunPlayer();
  });
}
```



```
});  
  
}
```

edited Jan 15 at 18:07



[Dmitriy Fialkovskiy](#)
1,803 6 18 29

answered Jan 15 at 16:55



[Sukhdevsinh Jadeja](#)
21 1 4