



TypeScript Interview Questions for beginners

📅 January 7, 2016 (<https://www.talkingdotnet.com/typescript-interview-questions/>) 👤 Talking Dotnet (<https://www.talkingdotnet.com/author/talkingdotnet/>) ➔ Interview Questions (<https://www.talkingdotnet.com/interview-questions/>), TypeScript (<https://www.talkingdotnet.com/typescript/>)

Previously I posted about [Grunt](https://www.talkingdotnet.com/grunt-js-interview-questions/) (<https://www.talkingdotnet.com/grunt-js-interview-questions/>) and [Gulp interview questions](https://www.talkingdotnet.com/gulp-js-interview-questions/) (<https://www.talkingdotnet.com/gulp-js-interview-questions/>) and in this post, find list of latest and updated TypeScript interview questions and their answers for freshers as well as experienced users. These interview questions will help you to prepare for the interviews, for quick revision and provide strength to your technical skills.

Without JavaScript, web development can't be imagined. Though JavaScript has issues but it is universally supported by all browsers. And present time is an era of new client side frameworks or technologies (AngularJs, [Gulp](https://www.talkingdotnet.com/gulp-js-interview-questions/) (<https://www.talkingdotnet.com/gulp-js-interview-questions/>), [Grunt](https://www.talkingdotnet.com/grunt-js-interview-questions/) (<https://www.talkingdotnet.com/grunt-js-interview-questions/>), [Yeoman](https://www.talkingdotnet.com/why-yeoman-for-asp-net-5-) (<https://www.talkingdotnet.com/why-yeoman-for-asp-net-5->



applications/) etc.) for better user experience and web development. TypeScript (<http://www.typescriptlang.org/>) is another player and is becoming popular as it lets you write JavaScript the way you really want to and addresses JavaScript issues.

TypeScript interview questions

Recommended Reading

Gulp js Interview questions (<https://www.talkingdotnet.com/gulp-js-interview-questions/>)

Grunt js Interview questions (<https://www.talkingdotnet.com/grunt-js-interview-questions/>)

Q1. What is TypeScript and why do we need it?

Ans. JavaScript is the only client side language universally supported by all browsers. But JavaScript is not the best designed language. It's not a class-based object-oriented language, doesn't support class based inheritance, unreliable dynamic typing and lacks in compile time error checking. And TypeScript addresses all these problems. In other words, TypeScript is an attempt to "fix" JavaScript problems.

TypeScript is a free and open source programming language developed and maintained by Microsoft. It is a strict superset of JavaScript, and adds **optional static typing** and **class-based object-oriented programming** to the language. TypeScript is quite easy to learn and use for developers familiar with C#, Java and all strong typed languages. At the end of day "TypeScript is a language that generates plain JavaScript files."

As stated on TypeScript official website (<http://www.typescriptlang.org/>), "TypeScript lets you write JavaScript the way you really want to. TypeScript is a typed superset of JavaScript that compiles to plain JavaScript. Any browser. Any host. Any OS. Open Source." Where "**typed**" means that it considers the types of variables, parameters and functions.

Q2. What are the benefits of TypeScript?

Ans. TypeScript has following benefits.

- It helps in code structuring.
- Use class based object oriented programming.
- Impose coding guidelines.
- Offers type checking.
- Compile time error checking.
- Intellisense.

Q3. What are different components of TypeScript?

Ans. There are mainly 3 components of TypeScript .

1. **Language** – The most important part for developers is the new language. The language consist of new syntax, keywords and allows you to write TypeScript.
2. **Compiler** – The TypeScript compiler is open source, cross-platform and open specification, and is written in TypeScript. Compiler will compile your TypeScript into JavaScript. And it will also emit error, if any. It can also help in concating different files to single output file and in generating source maps.
3. **Language Service** – TypeScript language service which powers the interactive TypeScript experience in Visual Studio, [VS Code](https://www.talkingdotnet.com/what-is-visual-studio-code-and-difference-between-visual-studio-2015/) (<https://www.talkingdotnet.com/what-is-visual-studio-code-and-difference-between-visual-studio-2015/>), Sublime, the TypeScript playground and other editor.

Q4. How can you get TypeScript and install it?

Ans. TypeScript can be installed and managed via npm, the Node.js package manager. To install TypeScript, first ensure the npm is installed properly. And then run following command to install TypeScript globally on your system.

```
1 | npm install -g typescript
```

^

TypeScript is included in Visual Studio 2013 Update 2 and Visual Studio 2015 by default. TypeScript also provides support for other editors like [Visual Studio Code](https://www.talkingdotnet.com/what-is-visual-studio-code-and-difference-between-visual-studio-2015/) (<https://www.talkingdotnet.com/what-is-visual-studio-code-and-difference-between-visual-studio-2015/>), sublime, Emacs and Vim.

Q5. How do you compile TypeScript files?

Ans. The extension for any TypeScript file is **".ts"**. And any JavaScript file is TypeScript file as it is a superset of JavaScript. So change extension of **".js"** to **".ts"** file and your TypeScript file is ready. To compile any **.ts** file into **.js** use following command.

```
1 | tsc <TypeScript File Name>
```

For example, to compile "Helloworld.ts"

```
1 | tsc helloworld.ts
```

And the result would be helloworld.js

Q6. Is it possible to combine multiple .ts files into a single .js file?

Ans. Yes, it possible. While compiling add **--outFile [OutputJSFileName]** option.

```
1 | tsc --outFile comman.js file1.ts file2.ts file3.ts
```

This will compile all 3 **".ts"** file and output into single **"comman.js"** file. And what will happen if you don't provide a output file name.

```
1 | tsc --outFile file1.ts file2.ts file3.ts
```

In this case, file2.ts and file3.ts will be compiled and the output will be placed in file1.ts. So now your file1.ts contains JavaScript code.

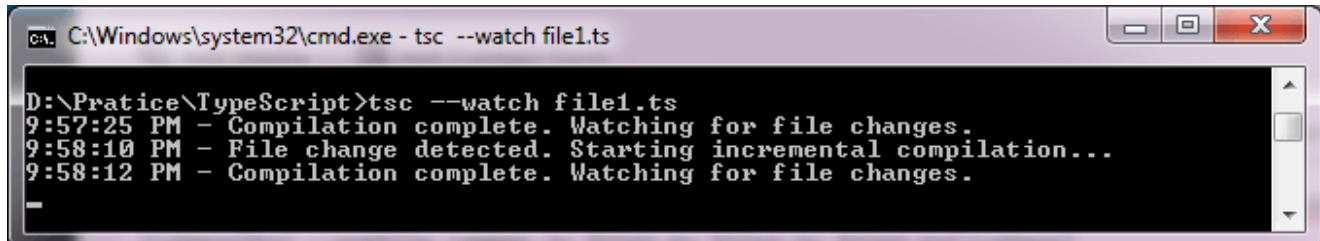
Q7. Is it possible to compile .ts automatically with real time changes in .ts file?

Ans. Yes. Using **--watch** compiler option this can be achieved.

^

```
1 | tsc --watch file1.ts
```

This will first compile file1.ts in file.js and watch for the file changes. As soon as there is any change detected, it will compile it again. But you need to ensure that command prompt must not be closed, used with `--watch` option.



(<https://www.talkingdotnet.com/wp-content/uploads/2015/12/TypeScript-Watch-option.png>)

Q8. Does TypeScript support all object oriented principles?

Ans. The answer is **YES**. There are 4 main principles to Object Oriented Programming: Encapsulation, Inheritance, Abstraction, and Polymorphism. TypeScript can implement all four of them with its smaller and cleaner syntax. Read [Write Object-Oriented JavaScript with TypeScript](http://rachelappel.com/write-object-oriented-javascript-with-typescript/) (<http://rachelappel.com/write-object-oriented-javascript-with-typescript/>).

Q9. Which object oriented terms are supported by TypeScript?

Ans. TypeScript supports following object oriented terms.

- Modules
- Classes
- Interfaces
- Data Types
- Member functions

Q10. Which are the different data types supported by TypeScript?

Ans. TypeScript supports following data types.

- Boolean `var bValue: boolean = false;`
- Number `var age: number = 16;`
- String `var name: string = "jon";`
- Array `var list:number[] = [1, 2, 3];`
- Enum

```
1 | enum Color {Red, Green, Blue};  
2 | var c: Color = Color.Green;
```

- Any `var unknownType: any = 4;`
- Void

```
1 | function NoReturnType(): void {  
2 | }
```

Q11. How TypeScript is optionally statically typed language?

Ans. TypeScript is referred as optionally statically typed, which means you can ask the compiler to ignore the type of a variable. Using `any` data type, we can assign any type of value to the variable. TypeScript will not give any error checking during compilation.

```
1 | var unknownType: any = 4;  
2 | unknownType = "Okay, I am a string";  
3 | unknownType = false; // A boolean.
```

Q12. What are modules in TypeScript?

Ans. Modules are the way to organize code in TypeScript. Modules don't have any features, but can contain classes and interfaces. It is same like namespace in C#.

Q13. What are classes in TypeScript?

Ans. The concept of classes is very similar to .Net/Java. A Class can have constructor, member variables, properties and methods. TypeScript also allows access modifiers "private" and "public" for member variables and functions.

^

Q14. How do you implement inheritance in TypeScript?

Ans. Using `extends` keyword, we can implement inheritance.

```
1  class Animal {  
2      public domestic:boolean;  
3      constructor(public name: string) { }  
4  }  
5  
6  class Cat extends Animal {  
7      constructor(name: string, domestic: boolean)  
8      {  
9          super(name);  
10         this.domestic = true;  
11     }  
12 }  
13  
14 class Tiger extends Animal {  
15     constructor(name: string, domestic: boolean)  
16     {  
17         super(name);  
18         this.domestic = false;  
19     }  
20 }
```

Q15. How to call base class constructor from child class in TypeScript?

Ans. Using `super()` , we can call base class constructor, as seen in above code.

Q16. What is the default access modifier for members of a class in TypeScript?

Ans. In TypeScript, each member of class is **public** by default.

Q17. How can you allow classes defined in a module to accessible outside of the module?

Ans. Classes defined in a module are available within the module. Outside the module you can't access them.



```
1 | module Vehicle {
2 |     class Car {
3 |         constructor (
4 |             public make: string,
5 |             public model: string) { }
6 |     }
7 |     var audiCar = new Car("Audi", "Q7");
8 | }
9 | // This won't work
10 | var fordCar = Vehicle.Car("Ford", "Figo");
```

As per above code, `fordCar` variable will give us compile time error. To make classes accessible outside module use `export` keyword for classes.

```
1 | module Vehicle {
2 |     export class Car {
3 |         constructor (
4 |             public make: string,
5 |             public model: string) { }
6 |     }
7 |     var audiCar = new Car("Audi", "Q7");
8 | }
9 | // This works now
10 | var fordCar = Vehicle.Car("Ford", "Figo");
```

Q18. How does TypeScript support optional parameters in function as in JavaScript every parameter is optional for a function?

Ans. In JavaScript, every parameter is considered optional. If no value is supplied, then it is treated as `undefined`. So while writing functions in TypeScript, we can make a parameter optional using the `"?"` after the parameter name.

```
1 | function Demo(arg1: number, arg2? :number) {
2 | }
```

So, `arg1` is always required and `arg2` is optional parameter. Remember, **Optional parameters must follow required parameters**. If we want to make `arg1` optional, instead of `arg2` then we need to change the order and `arg1` must be put after `arg2`.

```
1 | function Demo(arg2: number, arg1? :number) {
2 | }
```

Similar to optional parameters, default parameters are also supported.


```
1 | function Demo(arg1: number, arg2 = 4) {  
2 | }
```

Q19. Does TypeScript supports function overloading as JavaScript doesn't support function overloading?

Ans. Yes, TypeScript does support function overloading. But the implementation is odd.

When you overload in TypeScript you only have one implementation with multiple signatures.

```
1 | class Customer {  
2 |     name: string;  
3 |     Id: number;  
4 |  
5 |     add(Id: number);  
6 |     add(name:string);  
7 |     add(value: any) {  
8 |         if (value && typeof value == "number") {  
9 |             //Do something  
10 |         }  
11 |         if (value && typeof value == "string") {  
12 |             //Do Something  
13 |         }  
14 |     }  
15 | }
```

The first signature has one parameter of type number whereas the second signature has a parameter of type string. The third function contains the actual implementation and has a parameter of type any. The any data type can take any type of data. The implementation then checks for the type of the supplied parameter and execute different piece of code based on supplier parameter type.

OR You can also use union type introduced in TypeScript 1.4. Union types let you represent a value which is one of multiple types.

```
1 | add(a:string|number) {  
2 |     //do something  
3 | }
```

Using union type, you can typically remove the need for an overload.

Q20. Is it possible to debug any TypeScript file?

^

Ans. Yes, it is possible. To debug it, you need .js source map file. If you are new to source map, read more [here](#)

(<http://www.html5rocks.com/en/tutorials/developertools/sourcemaps/>). So compile the .ts file with the `--sourcemap` flag to generate a source map file.

```
1 | tsc --sourcemap file1.ts
```

This will create file1.js and file1.js.map. And last line of file1.js would be reference of source map file.

```
1 | ///  
//# sourceMappingURL=file1.js.map
```

Q21. What is TypeScript Definition Manager and why do you need it?

Ans. TypeScript Definition Manager (TSD) is a package manager to search and install TypeScript definition files directly from the community driven [DefinitelyTyped](http://definitelytyped.org/) (<http://definitelytyped.org/>) repository. Let's see with an example.

Suppose, you want to use some jQuery code in your .ts file.

```
1 | $(document).ready(function() { //Your jQuery code });
```

And now when you try to compile it using `tsc`, you will get compile time error **Cannot find name "\$**". That's because TypeScript can't understand what does "\$" means. So somehow we need to inform TypeScript compiler that it belongs to jQuery. That's where TSD comes into play. You can download jQuery Type Definition file and include it in your .ts file. First, install TSD.

```
1 | npm install tsd -g
```

In your typescript directory, create a new typescript project by running:

```
1 | tsd init
```

Then install the definition file for jquery.

```
1 | tsd query jquery --action install
```

This will download and create a new directory containing your jquery definition file. The definition file ends with “.d.ts”. So now include it by updating your typescript file to point to the jquery definition.

```
1 | /// <reference path="typings/jquery/jquery.d.ts" />  
2 | $(document).ready(function() { //To Do  
3 | });
```

Now try to compile again and this time js will be generated without any error.

So TSD will help you to get type definition file for required framework. If you wish to use angular, then download angular type definition file. You can find the complete list [here](http://definitelytyped.org/tsd/) (<http://definitelytyped.org/tsd/>).

Q22. What is TypeScript Declare Keyword?

Ans. It's quite possible that JavaScript libraries/frameworks don't have TypeScript definition files and yet you want to use them without any errors. The solution is to use the `declare` keyword. The `declare` keyword is used for **ambient declarations** where you want to define a variable that may not have originated from a TypeScript file.

```
1 | declare var unknownLibrary;
```

TypeScript runtime will assign `unknownLibrary` variable `any` type. You won't get Intellisense in design time but you will be able to use the library in your code.

Q23. How to generate TypeScript definition file from any .ts file?

Ans. You can generate TypeScript definition file from any .ts file via `tsc` compiler. Generating a TypeScript definition will make your TypeScript file reusable.

```
1 | tsc --declaration file1.ts
```

Q24. What is tsconfig.json file?

^

Ans. The presence of a `tsconfig.json` file in a directory indicates that the directory is the root of a TypeScript project. The `tsconfig.json` file specifies the root files and the compiler options required to compile the project. And using this file we can streamline building TypeScript project. Below is a sample `tsconfig.json` file.

```
1  {  
2    "compilerOptions": {  
3      "removeComments": true,  
4      "sourceMap": true  
5    },  
6    "files": [  
7      "main.ts",  
8      "othermodule.ts"  
9    ]  
10 }
```

Within files section, define all the `.ts` files in the project. And When invoke `tsc` without any other arguments with the above file in the current directory, it will compile all the files with the given compiler option settings.

Q25. What are disadvantages of TypeScript?

Ans. Well, TypeScript is great but there are some disadvantage as well.

- TypeScript is just another way to write JavaScript. It doesn't fix the problems of JavaScript. It just creates an illusion that it does.
- One more tool to learn.
- TypeScript has dependency on type definition files, if you wish to use any existing JavaScript libraries.
- Quality of type definition files is a concern as how can you be sure the definitions are correct?
- Frequent releases of new versions JavaScript library is also a pain area. Because if their type definition files are not updated then you can't use them instantly.
- In order to run the application in the browser, a compile step is required to transform TypeScript into JavaScript.

^

- Web developers are using JavaScript from decades and TypeScript doesn't bring anything new.
- To use any third party library, definition file is you need. And not all the third party library have definition file available.

That's all folk. If you have something to add in this list, please mention in comments section. Meanwhile, share this in your network and spread this to help others.


PS: If you found this content valuable and want to return the favour, then


 (<https://www.buymeacoffee.com/talkingdotnet>)

Spread this:


 (<https://www.talkingdotnet.com/typescript-interview-questions/?share=email&nb=1>)

 (<https://www.talkingdotnet.com/typescript-interview-questions/#print>)

 (<https://www.talkingdotnet.com/typescript-interview-questions/?share=facebook&nb=1>)


 (<https://www.talkingdotnet.com/typescript-interview-questions/?share=twitter&nb=1>)

 (<https://www.talkingdotnet.com/typescript-interview-questions/?share=telegram&nb=1>)

 (<https://www.talkingdotnet.com/typescript-interview-questions/?share=pinterest&nb=1>)

 (<https://www.talkingdotnet.com/typescript-interview-questions/?share=linkedin&nb=1>)

 (<https://www.talkingdotnet.com/typescript-interview-questions/?share=skype&nb=1>)

 (whatsapp://send?text=TypeScript%20Interview%20Questions%20for%20beginners
https%3A%2F%2Fwww.talkingdotnet.com%2Ftypescript-interview-questions%2F)

 (https://www.talkingdotnet.com/typescript-interview-questions/?share=reddit&nb=1)

 (https://www.talkingdotnet.com/typescript-interview-questions/?share=pocket&nb=1)

 (https://www.talkingdotnet.com/typescript-interview-questions/?share=tumblr&nb=1)

 More

Related

Gulp js interview questions for beginners

(https://www.talkingdotnet.com/gulp-js-interview-questions/)
In "Gulp"

Grunt js interview questions for beginners

(https://www.talkingdotnet.com/grunt-js-interview-questions/)
In "Grunt"

Tools for bundling and minification in ASP.NET

Core
(https://www.talkingdotnet.com/tools-for-bundling-minification-aspnet-core/)
In "ASP.NET Core"

Interview Questions (https://www.talkingdotnet.com/tag/interview-questions/)

TypeScript (https://www.talkingdotnet.com/tag/typescript/)

One thought on “TypeScript Interview Questions for beginners”

ALEX (HTTP://WWW.FULLSTACK.CAFE)

August 1, 2018 at 7:20 PM (https://www.talkingdotnet.com/typescript-interview-questions/#comment-44155)

Thanks for the great post! Your interview questions and answers have been added to https://www.fullstack.cafe (https://www.fullstack.cafe) and back-linked! Also TSD now (2018) is deprecated.

REPLY (HTTPS://WWW.TALKINGDOTNET.COM/TYPESCRIPT-INTERVIEW-QUESTIONS/?REPLYTOCOM=44155#RESPOND)

LEAVE A REPLY

Your email address will not be published. Required fields are marked *

Comment



Name *

Email *

Website

☐ Notify me of follow-up comments by email.☐ Notify me of new posts by email.

◀ Quick summary of what's changed in ASP.NET Core (<https://www.talkingdotnet.com/summary-what-changed-in-aspnet5/>)

What's coming in ASP.NET 5 RC 2 ▶ (<https://www.talkingdotnet.com/whats-coming-in-asp-net-5-rc-2/>)




<https://www.buymeacoffee.com/talkingdotnet>

GET UPDATES IN YOUR INBOX

FOLLOW US



(htt (htt
(htt  ps:/ ps:/
ps:/ (htt /ww /ww
/ww ps:/ w.fa w.pi
w.ta /twi ceb nter
lkin tter. ook. est.
gdo co co co
tnet m/t m/t m/t
.co alki alki alki
m/f ngd ngd ngd
eed otn otn otn
/) et) et) et/)

Online Registrations 2019

Uttanchal University

Engineering, Management, Mass
Communication, Agriculture & App
Science,

[APPLY NOW](#)



RECENT POSTS

Add Authentication to Angular 7 App using ASP.NET Core 3 (<https://www.talkingdotnet.com/add-authentication-to-angular-7-app-using-asp-net-core-3/>)

ASP.NET Core 3.0 App with .NET Core 3.0 preview 2 release (<https://www.talkingdotnet.com/asp-net-core-3-0-app-with-net-core-3-preview-2-and-visual-studio-2019/>)

Visual Studio 2019 Preview 2 release new features (<https://www.talkingdotnet.com/visual-studio-2019-preview-2-release-update/>)

How You Can Create a .NET Core Application Using Entity Framework Core with Oracle (<https://www.talkingdotnet.com/create-net-core-application-using-entity-framework-core-with-oracle/>)

7 Simple Hacks to Supercharge Your Angular 7 App (<https://www.talkingdotnet.com/7-simple-hacks-supercharge-web-applications-angular-7/>)

Creating First ASP.NET Core 3.0 App with Visual Studio 2019 Preview (<https://www.talkingdotnet.com/creating-first-asp-net-core-3-0-app-visual-studio-2019/>)

How to set Canonical URL in Angular 7 (<https://www.talkingdotnet.com/how-to-set-canonical-url-in-angular-7/>)



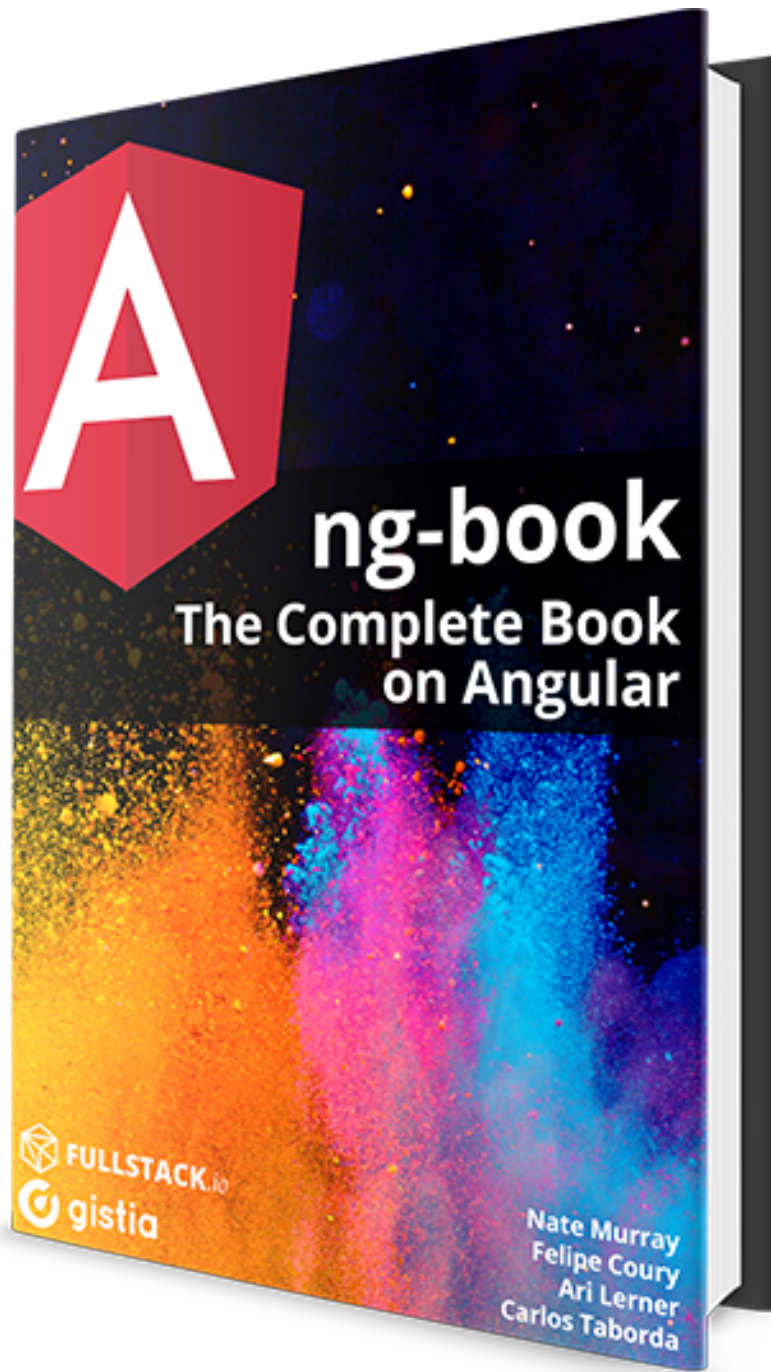


LEARN REACT THE RIGHT WAY

The up-to-date, in-depth,
complete guide to React and
friends.

(<https://gumroad.com/a/391689331>)

^



(<https://gumroad.com/a/324842611>)

POPULAR POSTS

- (<https://www.talkingdotnet.com/how-to-create-an-angular-6-app-with-visual-studio-2017/>)
How to create an Angular 6 app with Visual Studio 2017
24 May , 2018
- (<https://www.talkingdotnet.com/upgrade-angular-5-app-angular-6-visual-studio-2017/>)
Upgrade Angular 5 app to Angular 6 with Visual Studio 2017
08 May , 2018

(<https://www.talkingdotnet.com/Implement-ASP.NET-Core-SPA-template-features-in-an-Angular-6-app/>)
29 May , 2018



Upgrade Angular 4 app to Angular 5 with Visual Studio 2017
(<https://www.talkingdotnet.com/upgrade-angular-4-app-angular-5-visual-studio-2017/>)
20 Dec , 2017



Difference between Angular 1.x and Angular 2 (<https://www.talkingdotnet.com/difference-between-angular-1-x-and-angular-2/>)
14 Mar , 2016

(<https://www.talkingdotnet.com/Import-and-Export-excel-in-ASP.NET-Core-2.0-Razor-Pages/>)
20 Nov , 2017



Global Exception Handling in ASP.NET Core WEB API
(<https://www.talkingdotnet.com/global-exception-handling-in-aspnet-core-webapi/>)
17 Aug , 2016

(<https://www.talkingdotnet.com/Handle-Ajax-Requests-in-ASP.NET-Core-Razor-Pages/>)
30 Oct , 2017

CATEGORIES

.NET 5.4	.NET CORE	ANGULAR	ANGULAR 2	ASP.NET	ASP.NET 5	ASP.NET CORE
ASP.NET MVC	BLAZOR	ENTITY FRAMEWORK	GRUNT	GULP	INTERVIEW QUESTIONS	
MVC 6	NANCYFX	REACT	SQL SERVER	TESTING	TYPESCRIPT	VISUAL STUDIO
WEB API	YEOMAN					

RECENT POSTS

Add Authentication to Angular 7 App using ASP.NET Core 3 (<https://www.talkingdotnet.com/add-authentication-to-angular-7-app-using-asp-net-core-3/>)

ASP.NET Core 3.0 App with .NET Core 3.0 preview 2 release (<https://www.talkingdotnet.com/asp-net-core-3-0-app-with-net-core-3-preview-2-and-visual-studio-2019/>)

Visual Studio 2019 Preview 2 release new features (<https://www.talkingdotnet.com/visual-studio-2019-preview-2-release-update/>)

How You Can Create a .NET Core Application Using Entity Framework Core with Oracle (<https://www.talkingdotnet.com/create-net-core-application-using-entity-framework-core-with-oracle/>)

7 Simple Hacks to Supercharge Your Angular 7 App (<https://www.talkingdotnet.com/7-simple-hacks-supercharge-web-applications-angular-7/>)

JOIN THE NEWSLETTER

Enter your email address to subscribe to this blog and receive notifications of new posts by email.



Contact (<https://www.talkingdotnet.com/contact-us/>)

Privacy Policy (<https://www.talkingdotnet.com/privacy-policy/>)

FOLLOW US





 (htt (htt (htt
 (htt  ps:/ ps:/ ps:/
 ps:/ (htt /ww /plu /ww
 /ww ps:/ w.fa s.go w.pi
 w.ta /twi ceb ogle nter
 lkin tter. ook. .co est.
 gdo co co m/+ co
 tnet m/t m/t Talk m/t
 .co alki alki ing alki
 m/f ngd ngd dot ngd
 eed otn otn net/ otn
 /) et) et)) et/)