**TS Dosyalarini Compile Etmek**

tsc app.ts --watch veya tsc app.ts -w

**Birden fazla ts dosyası varsa**

tsc --init yapıyoruz ve tsconfig.json dosyasını alıyoruz

Bundan sonra tsc yaparak tüm ts dosyalarını js dosyasına compile ederiz.

tsc -w veya tsc --watch ile tüm dosyaları izleyerek çeviririz.

compile edilmesini istemediğimiz dosyaları tsconfig içerisine bu şekilde ekleriz

  "exclude": [

      "analytics.ts"

  ],

Herhangi bir isimle başlayıp .ts ile biten hiçbirşey compile edilmez

  "exclude": [

      "\*.ts"

  ],

Herhangi bir dizinde ki herhangi bir isimle başlayan .ts compile edilmez

  "exclude": [

      "\*\*/\*.dev.ts"

  ],

node\_modules içinde ki hiçbir yüklenen modul compile edilmez.

node\_modules u eklemeye gerek yok default olarak zaten ekli ve compile edilmiyor

  "exclude": [

      "node\_modules"

  ],

Aynı uzantı şekilleriyle include işlemlerimizi de yapabiliriz

Boş bıraktıgımızda tsc yaparsak herşeyin include oldugunu düşenerek herşeyi compile eder

Yada aşağıdaki gibi veya yukarıdaki gibi yol isimleriyle sadece include edilmesi gereken yolları belirleyebilirsiniz

  "include": [

      "app.ts",

      "analytics.ts"

  ],

Buda dosya isimiyle include edbilmemize yarar. Sadece dosyalar compile edilir.

  "files": [

      "app.ts",

      "analytics.ts"

  ],

  "compilerOptions": {

***Hangi teknoloji ile compile edilmesini seçeriz. Default olan es5 iyidir.***

    "target": "es5",

*/\* Specify ECMAScript target version: 'ES3' (default), 'ES5', 'ES2015', 'ES2016', 'ES2017', 'ES2018', 'ES2019', 'ES2020', or 'ESNEXT'. \*/*

***module optional. Birden fazla dosya gibi include işlemlerini yapıyor.***

    "module": "commonjs",

*/\* Specify module code generation: 'none', 'commonjs', 'amd', 'system', 'umd', 'es2015', 'es2020', or 'ESNext'. \*/*

***lib içine default olarak tanıması gereken libraryleri yazarız.***

*"lib": [*

*"es6",*

*"dom",*

*"dom.iterable"*

*],*

*/\* Specify library files to be included in the compilation. \*/*

***AllowJS dosya .ts ile bitmese .js ile bitse bile compile etmemize yarıyor.***

*"allowJs": true,*

*/\* Allow javascript files to be compiled. \*/*

***Syntaxleri check yani kontrol eder***

*"checkJs": true,*

*/\* Report errors in .js files. \*/*

***Typescripti React ile kullanırken aktif ederiz aşağıdaki şekilde***

*"jsx": "preserve",*

*/\* Specify JSX code generation: 'preserve', 'react-native', or 'react'. \*/*

***Library yapımında projeyi diğer insanlara gönderirken kullanılıyor***

*"declaration": true,*

*/\* Generates corresponding '.d.ts' file. \*/*

*"declarationMap": true,*

*/\* Generates a sourcemap for each corresponding '.d.ts' file. \*/*

***Ekstra olarak tarayıcıda kaynaklarda .ts dosyasını bize gösterir. Bunun için .ts i compile ederken ektra .js.map dosyasınıda oluşturur.***

*"sourceMap": true,*

*/\* Generates corresponding '.map' file. \*/*

*"outFile": "./",*

*/\* Concatenate and emit output to single file. \*/*

***Compile edilecek dosyaların nereye compile edileceğini yazarız. Genelde root dizininde dist klasörüne compile yapılır.***

    "outDir": "./dist",

*/\* Redirect output structure to the directory. \*/*

***rootdir yani compile edilecek dosyaların nereden geldiği yazılır.***

***Ana dizinde yani rootda User klasörü ve içinde .ts olursa onuda kontrol eder ve klasör olarak diste compile eder.***

    "rootDir": "./src",

*/\* Specify the root directory of input files. Use to control the output directory structure with --outDir. \*/*

*"composite": true,*

*/\* Enable project compilation \*/*

*"tsBuildInfoFile": "./",*

*/\* Specify file to store incremental compilation information \*/*

***Yorumları compile etmez***

*"removeComments": true,*

*/\* Do not emit comments to output. \*/*

***Dosyalarımızı control eder ve potensiyel errorları bize gösterir output edilmiş dosyalar hariç***

*"noEmit": true,*

*/\* Do not emit outputs. \*/*

***Önemli değil***

*"importHelpers": true,*

*/\* Import emit helpers from 'tslib'. \*/*

***Kodlarımız hata verirse loop olursa sadece o zaman çalıştırılmalı***

*"downlevelIteration": true,*

*/\* Provide full support for iterables in 'for-of', spread, and destructuring when targeting 'ES5' or 'ES3'. \*/*

*"isolatedModules": true,*

*/\* Transpile each file as a separate module (similar to 'ts.transpileModule'). \*/*

***Default olarak false gelir.***

***Eger true olursa herhangi bir yerde, herhangi bir dosyada hata olursa hiç bir dosyayı compile etmez***

***Fakat false iken hata olsada tüm dosyaları hatalı hatasız compile eder***

    "noEmitOnError": false,

*/\* Strict Type-Checking Options \*/*

***Eger strict true ise altındaki diğer tüm kodlar true demekdir.***

***Yani strict = 7 satir kod demektir. Yani ya asagidaki 7 taneyi ekleyin ya sadece stricti ekleyin***

    "strict": true,

*/\* Enable all strict type-checking options. \*/*

***any type gerektiren hatalari gosterir. Yani bir fonksiyon icinde ki parametre ye eger bir deger verilmezse hata aliriz bu true oldugu icin. Yani bize any type gerektiren yerlerdeki hatalari gosterir.***

*"noImplicitAny": true,*

*/\* Raise error on expressions and declarations with an implied 'any' type. \*/*

***Typescript bilmedigi degerler icin null doner. Bazen bu nullari ! isareti ile sonlandiririz cunku biliriz ki o unlem isareti Javascript e compile edildikten sonra olmayacaktir cunku js dosyasi html e gidip atadigimiz degeri gordugu icin. Eger burayi false yaparsak bu unlemi koymamiza gerek kalmayacaktir ve bu tarz null uyarilarini bize gostermeyecektir.***

***Yani bir nevi true iken unlem isareti koydugumuz seyleri bildigimizi soyluyoir ve bu tarz null hatalarini bilerek yok sayiyoruz.***

*"strictNullChecks": true,*

*/\* Enable strict null checks. \*/*

*Ingilizce Aciklamaya Bak*

*"strictFunctionTypes": true,*

*/\* Enable strict checking of function types. \*/*

*Fonksiyonlara bind verebilmemizi saglar. Bu sayede bir fonksiyona .bind yapip null gonderebiliriz. Tabi fonksiyon string istiyorsa hata verecektir. Burayi false yaparak hatayi kapatabiliriz veya 2. parametre olarak string gondererek hatayi yine asabiliriz.*

*const add (value: string) => return value; => Gonderdigimiz add.bind(null, `Deneme`)*

*"strictBindCallApply": true,*

*/\* Enable strict 'bind', 'call', and 'apply' methods on functions. \*/*

*"strictPropertyInitialization": true,*

*/\* Enable strict checking of property initialization in classes. \*/*

*"noImplicitThis": true,*

*/\* Raise error on 'this' expressions with an implied 'any' type. \*/*

***compile edilmis dosyalara use strict ekleyerek denetimi gerceklestirir***

*"alwaysStrict": true,*

*/\* Parse in strict mode and emit "use strict" for each source file. \*/*

***Tanimlanmsi ama kullanilmamis parametreleri ve local variableslari yani ayni dosya icinde tanimlanan degerleri gosterir. Ve compile etmeye calisirsak bu sekilde hata aliriz eger aktifse. import edilenler global variableslardir.***

*"noUnusedLocals": true,*

*/\* Report errors on unused locals. \*/*

*Buda yine parametreler icin olanidir ornekgin fonksiyonda parantez icindeki bir parametrenin fonksiyon icinde kullanilmamasi gibi*

*"noUnusedParameters": true,*

*/\* Report errors on unused parameters. \*/*

*Hicbirsey donmeyen return yapmayan fonksiyonlarda hata gostermemize saglar*

*"noImplicitReturns": true,*

*/\* Report error when not all code paths in function return a value. \*/*

*"noFallthroughCasesInSwitch": true,*

*/\* Report errors for fallthrough cases in switch statement. \*/*

***Advance ayarlar Cogu projede bizi ilgilendirmeyecek ayarlar***

*/\* Module Resolution Options \*/*

*"moduleResolution": "node",*

*/\* Specify module resolution strategy: 'node' (Node.js) or 'classic' (TypeScript pre-1.6). \*/*

*"baseUrl": "./",*

*/\* Base directory to resolve non-absolute module names. \*/*

*"paths": {},*

*/\* A series of entries which re-map imports to lookup locations relative to the 'baseUrl'. \*/*

*"rootDirs": [],*

*/\* List of root folders whose combined content represents the structure of the project at runtime. \*/*

*"typeRoots": [],*

*/\* List of folders to include type definitions from. \*/*

*"types": [],*

*/\* Type declaration files to be included in compilation. \*/*

*"allowSyntheticDefaultImports": true,*

*/\* Allow default imports from modules with no default export. This does not affect code emit, just typechecking. \*/*

    "esModuleInterop": true,

*/\* Enables emit interoperability between CommonJS and ES Modules via creation of namespace objects for all imports. Implies 'allowSyntheticDefaultImports'. \*/*

*"preserveSymlinks": true,*

*/\* Do not resolve the real path of symlinks. \*/*

*"allowUmdGlobalAccess": true,*

*/\* Allow accessing UMD globals from modules. \*/*

***Default ayarlari iyidir degistirmeye cok gerek yok***

*/\* Source Map Options \*/*

*"sourceRoot": "",*

*/\* Specify the location where debugger should locate TypeScript files instead of source locations. \*/*

*"mapRoot": "",*

*/\* Specify the location where debugger should locate map files instead of generated locations. \*/*

*"inlineSourceMap": true,*

*/\* Emit a single file with source maps instead of having a separate file. \*/*

*"inlineSources": true,*

*/\* Emit the source alongside the sourcemaps within a single file; requires '--inlineSourceMap' or '--sourceMap' to be set. \*/*

***Daha sonra bakilacak bu ayarlara***

*/\* Experimental Options \*/*

*"experimentalDecorators": true,*

*/\* Enables experimental support for ES7 decorators. \*/*

*"emitDecoratorMetadata": true,*

*/\* Enables experimental support for emitting type metadata for decorators. \*/*

*/\* Advanced Options \*/*

    "forceConsistentCasingInFileNames": true

*/\* Disallow inconsistently-cased references to the same file. \*/*