

## پیکربندی تایپ اسکریپت

تایپ اسکریپت زبان اصلی برنامه نویسی برنامه‌های Angular است. جاوا اسکریپت زیر مجموعه‌ی تایپ اسکریپت است که در آن برای تجهیز و حفاظت از نوع داده‌ها از زمان طراحی پشتیبانی می‌شود.

مرورگرها نمی‌توانند به صورت مستقیم تایپ اسکریپت را اجرا کنند. تایپ اسکریپت حتماً باید با استفاده از کامپایلر tsc به جاوا اسکریپت ترنسپایل شود که برای انجام این کار نیاز به پیکربندی دارد.

در این صفحه به برخی از جنبه‌های پیکربندی تایپ اسکریپت و محیط آن که برای برنامه نویسان Angular مهم هستند می‌پردازیم. همچنین به فایل‌های زیر به صورت مفصل خواهیم پرداخت:

- [tsconfig.json](#) – پیکربندی کامپایلر تایپ اسکریپت.
- [typings](#) – فایل‌های اعلانی تایپ اسکریپت.

### tsconfig.json

معمولاً هم زمان با این که کامپایلر در حال تولید فایل‌های جاوا اسکریپت است، فایل پیکربندی تایپ اسکریپت را به نام tsconfig.json به پروژه‌ی خود اضافه کنید تا کامپایلر را هدایت و راهنمایی کنید.

برای دریافت اطلاعات بیشتر در رابطه با tsconfig.json به صفحه‌ی رسمی تایپ اسکریپت مراجعه کنید.

در راهنمای نصب از tsconfig.json استفاده شده است:

### tsconfig.json

```
{
  "compilerOptions": {
    "target": "es5",
    "module": "commonjs",
    "moduleResolution": "node",
    "sourceMap": true,
    "emitDecoratorMetadata": true,
```

```
"experimentalDecorators": true,  
"lib": ["es2015", "dom"],  
"noImplicitAny": true,  
"suppressImplicitAnyIndexErrors": true  
}  
}
```

این فایل شامل آپشن ها و flag هایی است که وجودشان برای برنامه های Angular الزامی است.

## suppressImplicitAnyIndexErrors و noImplicitAny

برنامه نویسان تایپ اسکریپت بر سر true یا false بودن پرچم noImplicitAny اختلاف دارند. پاسخ صحیحی برای این مشکل وجود ندارد و شما در آینده می توانید این پرچم را تغییر دهید. اما تصمیم حال حاضر شما می تواند در پروژه های بزرگتر اثرگذار باشد. به همین دلیل بررسی آن خالی از لطف نیست.

اگر پرچم noImplicitAny (در حالت پیش فرض) false باشد، و کامپایلر نتواند نوع متغیر را بر اساس چگونگی استفاده از آن حدس بزند، در این صورت کامپایلر بدون سر و صدا نوع متغیر را بر روی any قرار می دهد. منظور از *implicit any* همین است.

در این آموزش مقدار پرچم noImplicitAny بر روی true تنظیم شده است. اگر این مقدار بر روی true تنظیم شود و کامپایلر تایپ اسکریپت نتواند نوع متغیر را حدس بزند، کامپایلر همچنان به تولید فایل های جاوا اسکریپت ادامه می دهد، در عین حال خطا نیز می دهد. بسیاری از برنامه نویسان با تجربه این تنظیمات صریح تر را ترجیح می دهند، زیرا در زمان کامپایل بررسی نوع متغیر منجر به درک خطاهای غیرعمدی بیشتری می شود.

می توانید نوع متغیر را بر روی any قرار دهید؛ حتی وقتی که مقدار پرچم noImplicitAny برابر با true باشد.

وقتی که مقدار پرچم noImplicitAny برابر با true است می توانید خطاهای ضمنی شاخص را نیز دریافت کنید. اغلب برنامه نویسان بر این باورند که این نوع از خطا بیشتر از آن که مفید باشد، آزاردهنده است. این خطاها را می توانید به کمک پرچم اضافی زیر محدود کنید:

```
"suppressImplicitAnyIndexErrors":true
```

در این آموزش این پرچم نیز بر روی true تنظیم شده است.

## تایپینگ های تایپ اسکریپت

بسیاری از کتابخانه های جاوا اسکریپت مانند جی کوئری، کتابخانه های آزمایشی Jasmine و Angular، محیط جاوا اسکریپت را به کمک ویژگی ها و سینتکس هایی که کامپایلر تایپ اسکریپت به صورت بومی آن ها را نمی شناسد، توسعه می دهند. زمانی که کامپایلری چیزی را نشناسد، خطا می دهد.

برای آن که کامپایلر را با کتابخانه هایی که بارگیری می کنید آشنا کنید، از فایل های تعریف نوع تایپ اسکریپت (d.ts files) استفاده کنید.

ویرایشگرهای آگاه به تایپ اسکریپت برای نمایش اطلاعات نوع مربوط به ویژگی های کتابخانه، تا حد امکان از همین فایل های تعریفی استفاده می کنند.

بسیاری از کتابخانه ها این فایل ها را در بسته های npm خود جای داده اند تا هم ویرایشگرها و هم کامپایلر تایپ اسکریپت بتوانند آن ها را پیدا کنند. Angular نیز چنین کتابخانه ای است. پوشه ای `node_modules/@angular/core/` مربوط به تمامی برنامه های Angular، شامل فایل های متعدد `d.ts` است، که این فایل های بخش های Angular را توصیف می کنند.

جهت دریافت فایل های تایپینگ برای بسته های کتابخانه ای خود که شامل فایل های `d.ts` هستند، نیازی نیست کاری انجام دهید. زیرا بسته های Angular به صورت پیش فرض دارای این فایل ها هستند.

## Lib.d.ts

تایپ اسکریپت شامل فایل اعلانی ویژه ای به نام `lib.d.ts` است. این فایل شامل اعلان های محیطی برای سازه های جاوا اسکریپت رایج موجود در زمان های اجرای جاوا اسکریپت و DOM است.

تایپ اسکریپت بر اساس `target` - اعلان های محیطی بیشتری را مانند Promise اضافه می کند. البته اگر هدف `es6` باشد.

با توجه به اینکه `es5` هدف QuickStart است، می توانید لیست فایل های اعلانی را به گونه ای لغو کنید که شامل کد زیر باشد:

```
"lib": ["es2015", "dom"]
```

به لطف مورد بالا، شما حالا تمامی تایپینگ های es6 را دارید، حتی اگر هدف es5 باشد.

## فایل‌های تایپینگ قابل نصب

بسیاری از کتابخانه‌ها از جمله جی کوئری، Jasmine و Lodash درون بسته‌های npm خود شامل فایل‌های d.ts نیستند.

خوشبختانه نویسندگان این کتابخانه‌ها و یا افراد فعال در انجمن آن‌ها، فایل‌های مجزای d.ts را برای این کتابخانه‌ها ایجاد کرده‌اند و آن‌ها را در مکان‌های معروفی منتشر کرده‌اند.

می‌توانید این تایپینگ‌ها را از طریق npm و با استفاده از بسته‌ی [@types/\\* scoped](#) نصب کنید و تایپ اسکریپت از ورژن 2 به بعد آن‌ها را به صورت خودکار شناسایی می‌کند.

برای نمونه، جهت نصب تایپینگ‌های مورد نیاز jasmine می‌توانید این کار را انجام دهید:

```
npm install @types/jasmine --save-dev
```

QuickStart دو تایپینگ یا فایل‌های d.ts زیر را می‌شناسد:

- تایپینگ‌های Jasmine برای فریمورک آزمایشی Jasmine.
- [node](#) برای کدهایی که به اشیاء موجود در محیط *Node.js* اشاره می‌کنند.

## Target یا هدف

به صورت پیش فرض، هدف es5 است. اما اگر می‌خواهید برنامه‌ی خود را تنها برای مرورگرهای سازگار با es6 توسعه دهید، می‌توانید هدف را بر روی es6 پیکربندی کنید. توجه داشته باشید اگر در یکی از مرورگرهای قدیمی مانند اینترنت اکسپلورر هدف را بر روی es6 پیکربندی کنید، با Syntax Error مواجه خواهید شد.