بسته‌های نرم افزاری Npm

Angular CLI ، برنامه‌های Angular و خود Angular به ویژگی‌ها و کارایی‌های ارائه شده توسط کتابخانه‌هایی وابسته هستند که این کتابخانه‌ها به صورت بسته‌های نرم افزاری npm موجود هستند.

برای دانلود و نصب این بسته‌های نرم افزاری می‌توانید از کلاینت npm کمک بگیرید. این کلاینت به صورت یک برنامه‌ی ®Node.js اجرا می‌شود.

کلاینت yarn می‌تواند جایگزین مناسبی برای دانلود و نصب این بسته‌ها باشد. زمانی که پروژه‌ی جدیدی را ایجاد می‌کنید، Angular CLI به صورت پیش فرض برای نصب این بسته‌ها از yarn استفاده می‌کند.

وجود Node.js و npm برای برنامه نویسی Angular ضروری است.

اگر تاکنون این دو را در سیستم خود نصب نکرده‌اید، همین الان آن‌ها را دریافت کنید.

حتماً مطمئن شوید که ورژن Node.js برابر با 8 و یا بیشتر و ورژن npm برابر با 5 و یا بیشتر باشد. برای بررسی این موضوع در پنجره‌ی کنسول و یا ترمینال دستورات node-v و npm-v را واردکنید. نسخه‌های قدیمی‌تر باعث بروز خطا می‌شوند.

برای مدیریت نسخه‌های مختلفی از Node.js و npm می‌توانید از nvm استفاده کنید. اگر همین الان پروژه‌هایی را در سیستم خود دارید که از ورژن های دیگری از Node.js و npm استفاده می‌کنند، باز هم nvm می‌تواند انتخاب مناسبی باشد.

package.json

npm و yarn هر دو بسته‌هایی را نصب می‌کنند که در یک فایل package.json شناخته شده هستند.

دستور ng new در CLI برای پروژه یتان یک فایل پیش فرض package.json را ایجاد می‌کند. این فایل مجموعه‌ای ابتدایی از بسته‌ها را مشخص می‌کند که به خوبی در کنار یکدیگر کار می‌کنند و به صورت مشترک از بسیاری از حالت‌های معمول برنامه‌ای پشتیبانی می‌کنند.

هم زمان با رشد برنامه‌تان شما باید بسته‌هایی را به package.json اضافه کنید. همچنین می‌توانید برخی از آن‌ها را حذف کنید.

در این آموزش به برخی از مهم‌ترین بسته‌های موجود در این مجموعه‌ی ابتدایی می‌پردازیم.

Dependencies و devDependencies

package.json در مجموعه‌ی خود شامل دو بسته‌ی dependencies و devDependencies است.

وجود dependencies برای اجرای برنامه ضروری است. این در حالی است که devDependencies تنها در توسعه‌ی برنامه کاربرد دارد.

Dependencies

بخش dependencies مربوط به package.json شامل موارد زیر است:

- بسته‌های Angular: هسته‌ی Angular و ماژول‌های اختیاری؛ اسم این بسته‌ها با /@angular آغاز می‌شود.
- بسته‌های پشتیبان: کتابخانه‌های سوم شخص که برنامه‌های Angular برای اجرا شدن به آن‌ها نیاز دارند.
- بسته‌های پلی فیل: پلی فیل‌ها شکاف‌هایی را در یکی از پیاده سازی های جاوا اسکریپت مرورگر وارد می‌کنند.

بسته‌های Angular

@angular/animations: کتابخانه‌ی انیمیشن‌های Angular تعریف و اعمال اثرات انیمیشنی مانند گذارهای لیستی و صفحه‌ای را آسان می‌کند. برای اطلاعات بیشتر به «آموزش انیمیشن‌ها» مراجعه کنید.

@angular/common: در این بسته سرویس‌ها، پایپ‌ها و دستورالعمل‌های پرکاربرد توسط تیم Angular ارائه شده‌اند. HttpClientModule نیز در همین بخش در زیر پوشه‌ی '@angular/common/http' قرار دارد.

@angular/core : اجزای حیاتی زمان اجرای فریمورک که همه‌ی برنامه‌ها به آن نیاز دارند. این بخش تمامی دکوراتورهای متادیتا، Component ، Directive ، تزریق وابستگی و هوک های چرخه‌ی عمر کامپوننت را شامل می‌شود.

@angular/compiler : کامپایلر قالب Angular. این کامپایلر قالب‌ها را درک می‌کند و آن‌ها را به کدی تبدیل می‌کند که برنامه‌ها به کمک آن بتوانند اجرا و رندر شوند. معمولاً نیازی به تعامل مستقیم با کامپایلر نیست؛ بلکه هنگام کامپایل کردن درجا در مرورگر از طریق platform-browser-dynamic از آن به صورت غیرمستقیم استفاده می‌شود.

@angular/forms : در این بخش از فرم‌های واکنشی و قالب محور پشتیبانی می‌شود.

@angular/http : کلاینت قدیمی و ناکارآمد Angular HTTP.

@angular/platform-browser : تمام چیزهای مربوط به DOM و مرورگر به ویژه بخش‌هایی که به رندر شدن المان‌ها در DOM کمک می‌کنند، در این بسته قرار دارند. همچنین در این بسته متد bootstrapModuleFactory() پیش بینی شده است تا بتوان برنامه‌هایی که در نسخه‌ی تولیدی قرار دارند و توسط AOT از قبل کامپایل شده‌اند را bootstrap کرد.

@angular/platform-browser-dynamic : شامل ارائه دهنده‌ها و متدهایی است که می‌توان به کمک آن‌ها و با استفاده از کامپایلر JIT برنامه را در کلاینت کامپایل و اجرا کرد.

@angular/router : بعد از آن که آدرس URL مرورگر تغییر کند، ماژول روتر بین صفحات برنامه‌ی شما به گشت و گذار می‌پردازد.

@angular/upgrade : مجموعه‌ای از ابزارها برای ارتقاء برنامه‌های AngularJS به Angular.

بسته‌های پلی فیل

بسیاری از مروگرها به صورت بومی از برخی از ویژگی‌های موجود در آخرین استانداردهای HTML پشتیبانی نمی‌کنند. با وجود این که Angular به این ویژگی‌ها نیاز دارد. پلی فیل‌ها می‌توانند کار این ویژگی‌ها را تقلید

کنند. در آموزش «پشتیبانی از مرورگر» توضیح داده شده است که کدام یک از مرورگرها به پلی فیل نیاز دارند و چگونه می‌توانید آن‌ها را اضافه کنید.

package.json پیش فرض بسته‌ی core-js را نصب می‌کند که در آن پلی فیل‌های مورد نیاز بسیاری از مرورگرهای محبوب وجود دارد.

بسته‌های پشتیبان

rxjs: بسیاری از API های observable Angular ها را برگشت می‌دهند. RxJS کاربردی از مشخصات observable ها می‌باشد که در حال حاضر نزد انجمن TC39 قرار دارد که کار آن تعیین استانداردهای لازم برای زبان جاوا اسکریپت است.

zone.js: بعد از آن که عملیات‌های بومی جاوا اسکریپت، رویدادها را ایجاد می‌کنند، Angular برای عملی شدن تغییر فرآیندهای شناسایی به Zone.js نیاز دارد. Zone.js کاربردی از مشخصاتی است که در حال حاضر نزد انجمن TC39 قرار دارد و کار آن تعیین استانداردهای لازم برای زبان جاوا اسکریپت است.

DevDependencies

بسته‌هایی که در بخش devDependencies فایل package.json قرار دارند، به شما در توسعه‌ی برنامه درون سیستم محلی خودتان کمک می‌کند.

بهتر است که این بسته‌ها در کنار برنامه‌ی تولیدی استفاده نشود. هرچند که ضرری در انجام این کار وجود ندارد.

@angular/cli : ابزارهای Angular CLI.

@angular/compiler-cli : کامپایلر Angular که توسط دستورات build و serve مربوط به Angular CLI احضار می‌شوند.

@angular/language-service : سرویس زبانی Angular قالب‌های کامپوننت را تحلیل می‌کند و اطلاعات مربوط به خطا و نوع را ارائه می‌کند. به گونه‌ای که ویرایشگرهای آگاه به تایپ اسکریپت بتوانند از آن‌ها استفاده

کنند تا تجربه‌ی برنامه نویس بهبود یابد. برای مثال، به «افزونه‌های سرویس زبانی Angular برای کد ویژوال استودیو» مراجعه کنید.

@types/...: فایل‌های تعریفی تایپ اسکریپت برای کتابخانه‌های سوم شخص مانند Jasmine و Node.js.

codelyzer: لینتر مورد نیاز برای برنامه‌های Angular که قوانین آن‌ها منطبق بر راهنمای ارائه شده برای استایل Angular است.

jasmine/...: بسته‌های مورد نیاز برای پشتیبانی از کتابخانه‌ی آزمایشی Jasmine.

karma/...: بسته‌های مورد نیاز برای پشتیبانی از اجرا کننده‌ی آزمایش karma.

protractor: یک چهارچوب e2e برای برنامه‌های Angular که تحت کنترل WebDriverJS ساخته شده است.

ts-node: محیط اجرای تایپ اسکریپت و REPL برای Node.js.

tslint: ابزار تحلیل استاتیک است که کار آن بررسی کد تایپ اسکریپت از نظر خوانا بودن، قابلیت نگهداری و خطاهای عملکردی است.

typescript: سرور زبانی تایپ اسکریپت از جمله کامپایلر tsc تایپ اسکریپت.

تعداد بسته‌ها و فایل‌ها بیش از حد زیاد است!

package.json پیش فرض، بسته‌های نرم افزاری را بیش از حد نیاز پروژه‌ی شما نصب می‌کند.

یک بسته‌ی مشخص ممکن است شامل ده‌ها، صدها، حتی هزاران فایل باشد و تمامی این فایل‌ها را در دایرکتوری node_modules سیستم محلی شما قرار دهد. حجم خالص فایل‌ها یک مقدار ترس برانگیز است.

شما می‌توانید بسته‌هایی که نیاز ندارید را حذف کنید اما چطور می‌توانید مطمئن شوید که دیگر به آن‌ها نیاز ندارید؟ یک راهکار عملی برای این کار این است که بسته‌ای که به آن نیاز ندارید را نصب کنید تا این که نگران حذف کردن آن باشید. فایل‌ها و بسته‌های اضافی موجود در سیستم برنامه نویسی محلی شما بی ضرر هستند.

به صورت پیش فرض Angular CLI بسته‌های نرم افزاری را داخل تنها یک فایل پردازش می‌کند، فقط همان تعداد معدودی از فایل‌های کتابخانه‌ای "vendor" که برنامه‌ی شما واقعاً به آن‌ها نیاز دارد. مرورگر این بسته‌ها را دانلود می‌کند (نه فایل‌های بسته‌های نرم افزاری اصلی را).

برای اطلاعات بیشتر به بخش «گسترش» مراجعه کنید.