آموزش شروع کار با Angular

Angular یک پلت فرم است که ساخت برنامه های وب را آسان می کند.

Angular ترکیبی از قالب های اخباری، Dependency Injection و بهترین شیوه های یکپارچه سازی برای حل چالش های توسعه است.

Angular توانایی برنامه نویسان را برای ساخت برنامه های کاربردی وب، موبایل یا دسکتاپ بالا می برد.

پیش نیازها

در این مقاله ما فرض را بر این می گذاریم که شما در حال حاضر با جاوا اسکریپت آشنا هستید و مفاهیم کلاس و ماژول را می دانید.

تمامی مثال های این مقاله با استفاده از Type Script نوشته می شود.

گام اول: نصب کردن محیط توسعه

Angular CLI یک ابزار رابط خط فرمان است که می تواند یک پروژه را ایجاد کند، فایل ها را اضافه کند و وظایف مختلفی مانند testing، bundling و deployment را انجام دهد.

قبل از اینکه بتوانید کاری انجام دهید، شما باید محیط توسعه خود را نصب کنید.

بنابراین در گام اول اگر Node.js و npm روی سیستم شما از قبل نصب نیست، می بایست ابتدا Node.js و npm را نصب نمایید.

برای نصب node.js شما می توانید از لینک https://nodejs.org/en/download آخرین نسخه آن را دانلود نمایید که با نصب آن npm هم نصب خواهد شد.

بعد از نصب می توانید با اجرای دستور node –v بر روی پنجره Command Prompt از نصب آخرین نسخه node.js که نسخه 8.x یا بالاتر می باشد اطمینان حاصل نمایید. همچنین شما می توانید با اجرای دستور npm –v نسخه npm نصبی بر روی سیستم خود را مشاهده نمایید که حداقل می بایست نسخه 5.x یا بالاتر باشد.

نسخه های قدیمی تر خطاهایی تولید می کنند، اما نسخه های جدیدتر خوب هستند و توصیه می شود که از نسخه های بالاتر تر استفاده نمایید.

سپس Angular CLI را با استفاده از دستور زیر به صورت Globally بر روی سیستم خود نصب نمایید.

```
npm install -g @angular/cli
```

گام دوم: ایجاد یک پروژه جدید

پنجره Command Prompt را باز کنید.

با اجرای دستور زیر، یک پروژه جدید و برنامه ی پیش فرض ایجاد کنید:

```
ng new my-app
```

Angular CLI پکیج های مورد نیاز npm را نصب می کند، فایل های پروژه را ایجاد می کند و پروژه را با یک برنامه پیش فرض ساده پر می کند.

این امر می تواند مدتی زمان ببرد.

شما می توانید با استفاده از دستور ng add یک pre-packaged را به یک پروژه جدید اضافه کنید.

گام سوم: راه اندازی Application

در این مرحله به دایرکتوری پروژه بروید و سرور را راه اندازی کنید.

```
cd my-app
ng serve --open
```

دستور ng serve سرور را راه اندازی می کند، فایل های خود را تماشا می کند و برنامه را دوباره بازسازی می کند، همچنان که شما آن فایل ها را تغییر می دهید.

با استفاده از گزینه open– یا (just -o) مرورگر شما به طور خودکار در http: // localhost: 4200 / باز خواهد شد.

برنامه شما با پیام، شما را خوش آمد می گوید:

گام چهارم: ویرایش اولین کامپوننت Angular

CLI اولین کامپوننت Angular را برای شما ایجاد کرد.

این کامپوننت، کامپوننت ریشه (root component) است و به نام app-root است.

شما می توانید آن را در src/app/app.component.ts/ پیدا کنید.

فایل component را باز کنید و عنوان آن را از «app» به « My First Angular App!» تغییر دهید.

src/app/app.component.ts

```
export class AppComponent {
  title = 'My First Angular App!';
}
```

browser به طور خودکار با عنوان جدید بارگذاری می شود. این خوب است، اما می تواند بهتر باشد.

فایل src/app/app.component.css را باز کنید و روی style آن کمی کار کنید.

src/app/app.component.css

```
h1 {
  color: #369;
  font-family: Arial, Helvetica, sans-serif;
  font-size: 250%;
}
```

تا اینجا از شما انتظار می رود که یک برنامه Hello World ساده را انجام دهید .

بررسی فایل پروژه

یک پروژه Angular CLI پایه ای برای آزمایش های سریع و راه حل های سازمانی است.

اولین فایلی که شما باید چک کنید فایل README.md است.

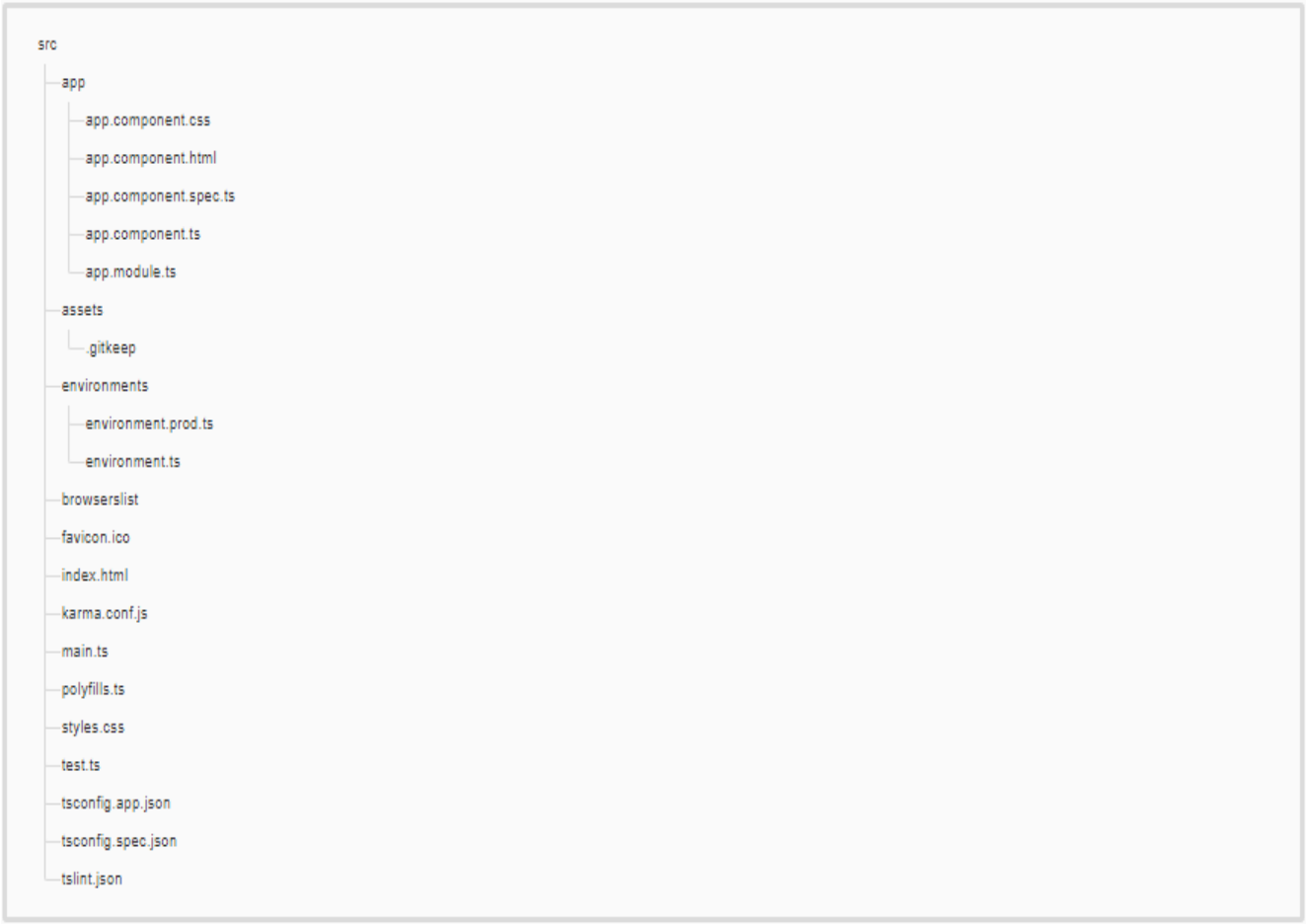
این فایل شامل بعضی از اطلاعات اولیه در مورد نحوه استفاده از دستورات CLI است.

برخی از فایل های ایجاد شده ممکن است برای شما نا آشنا باشد.

فولدر src

برنامه شما در پوشه src قرار می گیرد.

تمام کامپوننت های Angular، الگوها، سبک ها، تصاویر، و هر چیز دیگری که برنامه شما نیاز دارد، اینجا قرار دارد.

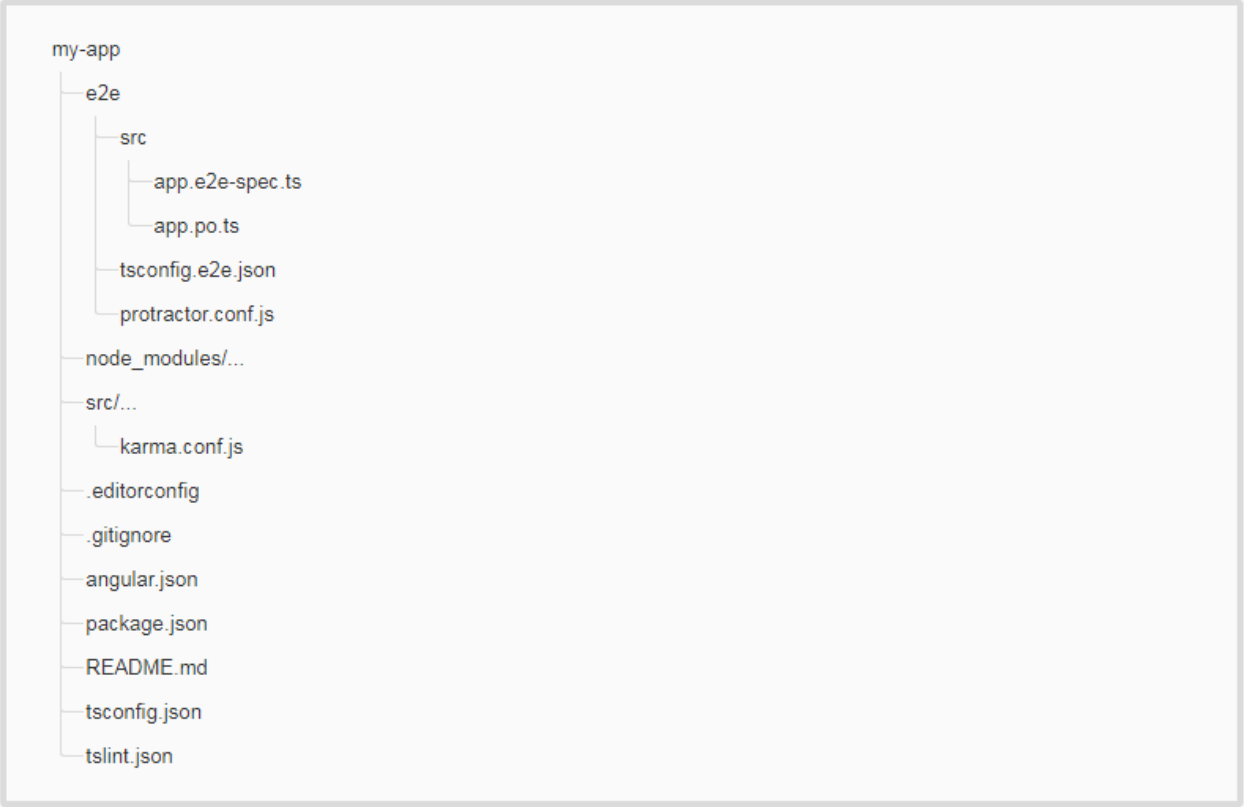


Purpose	File
AppComponent را همراه با یک قالب HTML، شیوه CSS و یک تست واحد تعریف می کند. این یک root component است که درختی از کامپوننت های تودرتو خواهد شد.	app/app.component.{ts,html,css,spec.ts}
AppModule را تعریف می کند، که یک root mudule است و به Angular می گوید که چگونه برنامه را اسمبل کند. در حال حاضر فقط AppComponent را اعلام می کند. به زودی کامپوننت های بیشتری برای اعلام اینجا خواهد گرفت.	app/app.module.ts
پوشه ای که در آن شما می توانید تصاویر و هر چیز دیگری را که در هنگام ساخت برنامه به آن نیاز دارید قرار دهید.	assets/*
این پوشه حاوی یک فایل برای هر یک از محیط های مقصد خود است.	environments/*

Browserslist	یک فایل پیکربندی برای به اشتراک گذاشتن مرورگرهای هدف بین ابزارهای مختلف.
favicon.ico	هر سایتی می خواهد در نوار bookmark خوب به نظر برسد. می توانید با آیکون Angular خود شروع کنید.
index.html	صفحه اصلی HTML که زمانی که کسی از سایت شما بازدید می کند ظاهر می گردد. اغلب اوقات شما هرگز نیازی به ویرایش آن ندارید. CLI به طور خودکار تمام فایل های js و css را هنگام ساختن برنامه خود اضافه می کند، بنابراین شما هرگز نباید هر گونه تگ <script> یا <link> را اینجا به صورت دستی اضافه کنید.
karma.conf.js	پیکربندی تست واحد برای karma test runner که در هنگام اجرای ng test مورد استفاده قرار می گیرد.
main.ts	نقطه ورود اصلی به برنامه است. برنامه را با کامپایلر JIT کامپایل می کند و AppModule (ماژول اصلی Application) را برای اجرا در مرورگر راه اندازی می کند. شما همچنین می توانید از کامپایلر AOT بدون تغییر هیچ کدی با اضافه کردن پرچم aot– به دستورات ng build یا ng serve استفاده کنید.
polyfills.ts	مرورگرهای مختلف سطوح مختلف پشتیبانی از استانداردهای وب را دارند. Polyfills به نرمال سازی این تفاوت ها کمک می کند. شما باید با core-js و zone.js کاملا امن باشید، اما برای اینکه مطمئن شوید برای اطلاعات بیشتر، راهنمای مرورگر را بررسی کنید.
styles.css	Style های سراسری شما اینجا هستند. اغلب اوقات شما می خواهید سبک های محلی را در کامپوننت های خود برای ویرایش سازی و نگهداری آسان تر داشته باشید، اما سبک هایی که بر کل برنامه های شما تأثیر می گذارد، باید در یک مکان مرکزی باشند.
test.ts	این نقطه ورودی اصلی برای Unit tests است. این فایل پیکربندی های سفارشی دارد که ممکن است نا آشنا باشد، اما چیزی نیست که شما باید ویرایش کنید.
tsconfig.{applspec}.json	پیکربندی کامپایلر TypeScript برای برنامه Angular (tsconfig.app.json) و برای آزمایش های واحد (tsconfig.spec.json).
tslint.json	پیکربندی اضافی Linting برای TSLint و Codelyzer که در هنگام اجرای ng lint استفاده می شود. Linting به شما کمک می کند تا سبک کد خود را حفظ کنید.

فولدر ریشه (root folder)

فولدر /src یکی از موارد درون پوشه ریشه پروژه است. فایل های دیگر به ساخت، تست، نگهداری، مستند سازی و استفاده از برنامه کمک می کنند. این فایل ها در پوشه ریشه در کنار /src قرار دارند.



File	Purpose
node_modules/	Node.js این پوشه را ایجاد می کند و تمام ماژول های شخص ثالث موجود در package.json را در داخل آن قرار می دهد.
.editorconfig	پیکربندی ساده برای ویرایشگر شما برای اطمینان از اینکه همه کسانی که از پروژه شما استفاده می کنند همان پیکربندی اساسی را دارند.
angular.json	پیکربندی برای Angular CLI است. در این فایل می توانید پیش فرض ها را تنظیم کنید و همچنین پیکربندی کنید که چه فایل هایی در هنگام ساخت پروژه شامل شوند.
package.json	تنظیمات npm را نشان می دهد. شما همچنین می توانید اسکریپت های سفارشی خود را در اینجا اضافه کنید.

README.md	داکیومنت اساسی برای پروژه شما
tsconfig.json	پیکربندی کامپایلر TypeScript برای IDE خود را انتخاب کنید.