# تعاریف

|  |  |  |  |
| --- | --- | --- | --- |
|  | -- | | |
| شماره سند: | شناسه سند | |  |
| طـبقه‌بندی: | عادی/حساس/محرمانه | |  |
| تــاریــخ: | روز/ماه/سال |  | |

v

SPA: اپلیکیشن تک صفحه ای Single Page Application

Directives: دستورالعمل‌ها (دایرکتیو) در انگولار یک کلاس است که به صورت @directive اعلان شده است.

# پیش نیازها

## محیط توسعه (IDE)

در این مستند از Visual Studio Code برای کد نویسی استفاده می کنیم.

## پلتفرم Node.JS

نودجی‌اس یک محیط اجرای جاوااسکریپتی است که روی موتور اجرای کد جاوااسکریپتِ مرورگر کروم تحت عنوان V8 ساخته شده است. برای نصب Packege ها از npm موجود در Node.JS استفاده می کنیم.

فایل نصبی آن را از سایت <https://nodejs.org/en> دانلود کنید و برای اطمینان از نصب صحیح Node.JS و npm از دو دستور زیر در cmd یا PowerShell ویندوز استفاده می‌کنیم.

|  |
| --- |
| node -v  npm -v |

## Angular CLI

Angular cli ابزاری است که به برنامه نویسان Angular کمک میکنند تا بتوانند بهتر و راحتتر با Angular کار کنند و تمرکز کامل خود را بر روی توسعه اپلیکشن خود قرار دهند و بخش های ایجاد فایل های مختلف، تست کدها، مشاهده خطاهای کاربری و ... رو به Cli واگذار کنند.

برای نصب Angular CLI از دستور زیر در cmd استفاده کنید.

|  |
| --- |
| npm install –g @angular/cli |

-g : این آپشن مشخص می کند که Angular CLI به صورت Global نصب شود. در صورتی که از پرچم -g استفاده نکنید فقط برای یک پروژه نصب می شود.

به علت مشکلات تحریم ممکن است برای نصب ابزار به پروکسی یا فیلترشکن نیاز داشته باشید.

# ایجاد پروژه جدید

برای ایجاد یک پروژه Angular جدید از کد زیر در cmd استفاده کنید. دقت کنید که فایل های پروژه در مسیری که اکنون قرار دارید ساخته خواهند شد برای اینکه در مسیر دلخواه شما ساخته شود از دستورات cmd برای رفتن به پوشه مورد نظر خود استفاده کنید.

|  |
| --- |
| ng new ProjectName |

پس از اجرای دستور فوق محتویات پروژه در فولدری با نام پروژه ای که وارد کرده اید ساخته می شود.

به علت مشکلات تحریم ممکن است به پروکسی یا فیلترشکن نیاز داشته باشید.

با زدن دستور زیر به پوشه پروژه وارد شوید.

|  |
| --- |
| cd ProjectName |

برای اجرای برنامه از دستور زیر استفاده کنید.

|  |
| --- |
| ng serve |

پس از اتمام کامپایل یک لینک به شما نمایش داده می شود که پورت اپلیکیشن در آن مشخص شده است. با وارد کردن لینک نمایش داده شده در مرورگر خود می توانید اپلیکیشن ساخته شده را مشاهده کنید.

# آشنایی با TypeScript

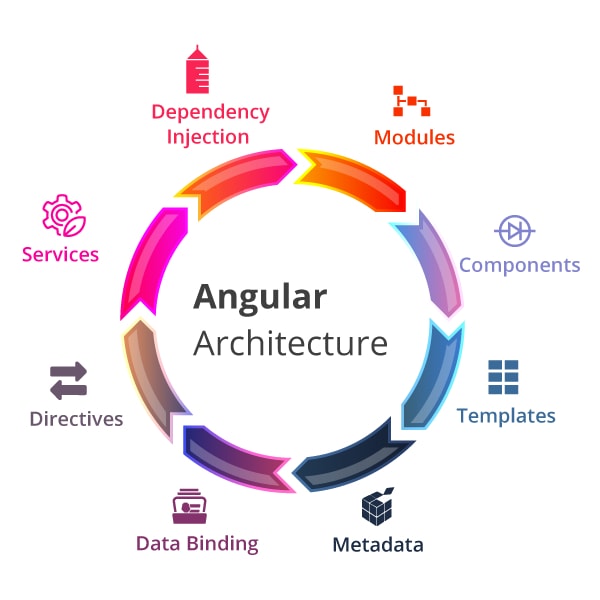
متن

# آشنایی با Webpack

متن

# ساختار پروژه انگولار

معماری انگولار دارای قابلیت‌های بسیاری است که به توسعه‌دهندگان برای ساخت برنامه کمک می‌کند.



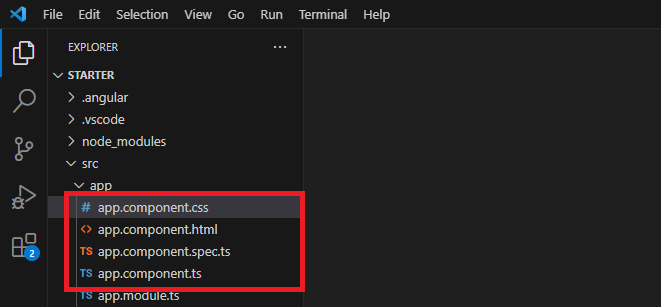
اساسا می‌توان گفت انگولار (Angular) یک چارچوب و همچنین بستری برای ساخت برنامه‌های مشتری (client) در HTML و TypeScript است. عملکرد انگولار توسط مجموعه از کتابخانه‌های TypeScript اداره می‌شود.

## اجزا (Components)

هر برنامه انگولار دارای حداقل یک کامپوننت است و به عنوان مؤلفه اصلی شناخته می‌شود که به همراه ایجاد یک پروژه جدید Angular ایجاد می شود که با عنوان کامپوننت اصلی یا ریشه نیز شناخته می شود.

کامپوننت از چهار فایل به شرح زیر تشکیل شده است:

* component.css: استایل های css مربوط به کامپوننت درون این فایل قرار می‌گیرد.
* component.html: کد های html مربوط به کامپوننت درون این فایل قرار می‌گیرد.
* component.spec.ts: این فایل مربوط به تست‌های برنامه نویسی است.
* component.ts: فایل اصلی کامپوننت است که شامل کلاس‌های تایپ اسکریپت است.



### ساخت کامپوننت با Angular CLI

برای ساخت یک کامپوننت در Angular می‌توانید از دستور زیر در Terminal ابزار Visual Studio Code یا هر IDE دیگری استفاده کنید.

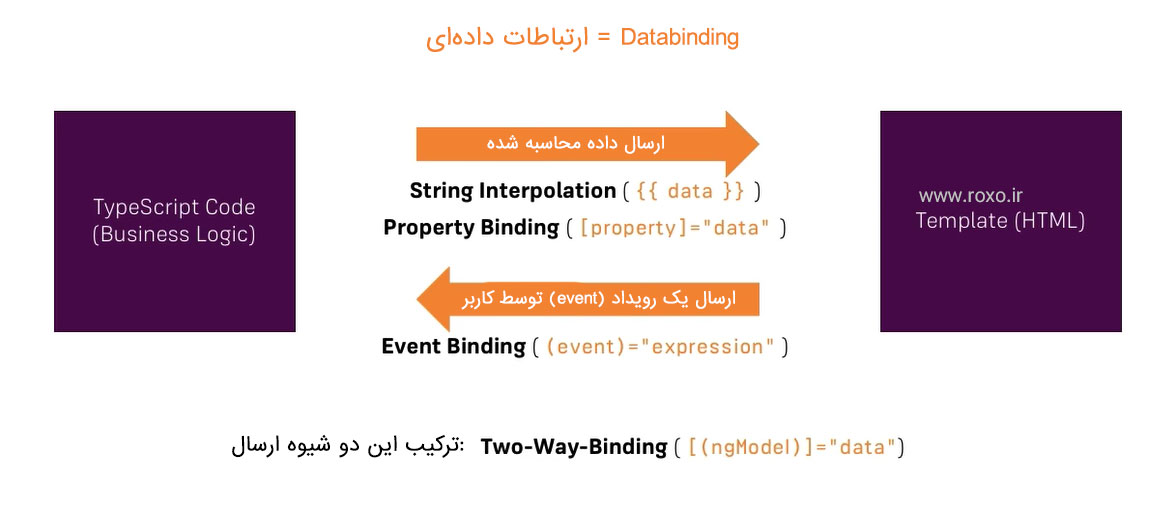
|  |
| --- |
| ng generate component about |

about: یک نام اختیاری برای کامپوننت است.

## Data Binding

در انگولار مفهومی تحت عنوان databing مطرح شد که به معنی ارتباطات داده‌ای بین قسمت منطق یا کنترلر برنامه که به زبان تایپ اسکریپت نوشته می‌شود با قسمت نمایشی برنامه که به زبان Html نوشته شده است.

به عبارت دیگر databinding به معنی ارتباط app.component.ts با app.component.html است. برای تفهیم بیشتر به تصویر زیر دقت کنید:



### نمایش داده با Interpolation

رایج‌ترین روش برای نمایش مقادیر اتصال‌یافته این است که آن‌ها را به صورت فرزندان عنصر DOM دیگر نمایش دهیم. ما می‌توانیم متغیر خود را با استفاده از نام آن از قالب‌ها و به وسیله دو آکولاد در طرفین مورد ارجاع قرار دهیم. به خاطر بسپارید که این روش تنها اتصال داده یک‌طرفه ارائه می‌کند:

|  |
| --- |
| <p>{{ first\_name }}</p> |

نکته: ابتدا باید یک متغییر در فایل component.ts با نام first\_name بسازید.

### Property binding

برخی عناصر HTML و کامپوننت‌های انگولار این الزام را دارند که مقادیر با استفاده از «خصوصیت‌ها» (attributes) به آن‌ها ارسال شود. این اتصال‌های داده به نام «اتصال‌های مشخصه» (property bindings) شناخته می‌شوند.

ساختار این روش نیز مشابه روش قبلی است و مانند این است که مقدار را به فرزندان عنصر الحاق می‌کنیم، تنها تفاوت در این روش آن است که از ساختاری به همراه خصوصیت‌ها استفاده می‌کنیم.

یکی از عناصری که مقدار را از طریق خصوصیت‌ها می‌گیرد، تگ input است:

|  |
| --- |
| <input id="first\_name" [value]="first\_name" /> |

### Event binding

تا به این جا تنها از انتقال داده برای نمایش مقادیر متغیرها از کنترلر به نما استفاده کرده‌ایم، اما اگر بخواهیم به اعمال انجام یافته از سوی کاربران واکنش نشان دهیم باید رویداد متناظر، برای نمونه رویداد کلیک را به یک متد اتصال دهیم که درون منطق کامپوننت اعلان شده است.

به این منظور از خصوصیت‌ها استفاده می‌کنیم و نام رویدادی که می‌خواهیم به آن اتصال پیدا کنیم را درون پرانتز می‌آوریم:

|  |
| --- |
| <button (click)="onButtonClicked()">Change First Name</button> |

در component.ts نیز متد خود را قرار می دهیم.

|  |
| --- |
| import { Component } from '@angular/core'  @Component({  selector: 'my-app',  templateUrl: './app.component.html',  styleUrls: ['./app.component.css'],  })  export class AppComponent {  first\_name = 'Angular'    onButtonClicked() {  this.first\_name = 'Charlie'  }  } |

### Class and style binding

برای اتصال داده ها از جزء به ویژگی کلاس HTML استفاده می شود.

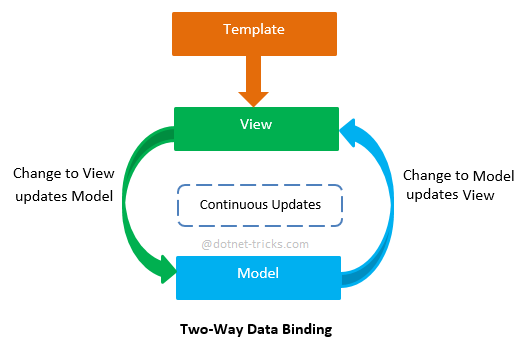
|  |
| --- |
| <p [class]="myClasses"> |

Style binding برای اتصال داده ها از جزء به ویژگی استایل HTML استفاده می شود.

|  |
| --- |
| <p [style.color]="myParaColor"> ... </p> |

### اتصال دو طرفه (Two-way binding)

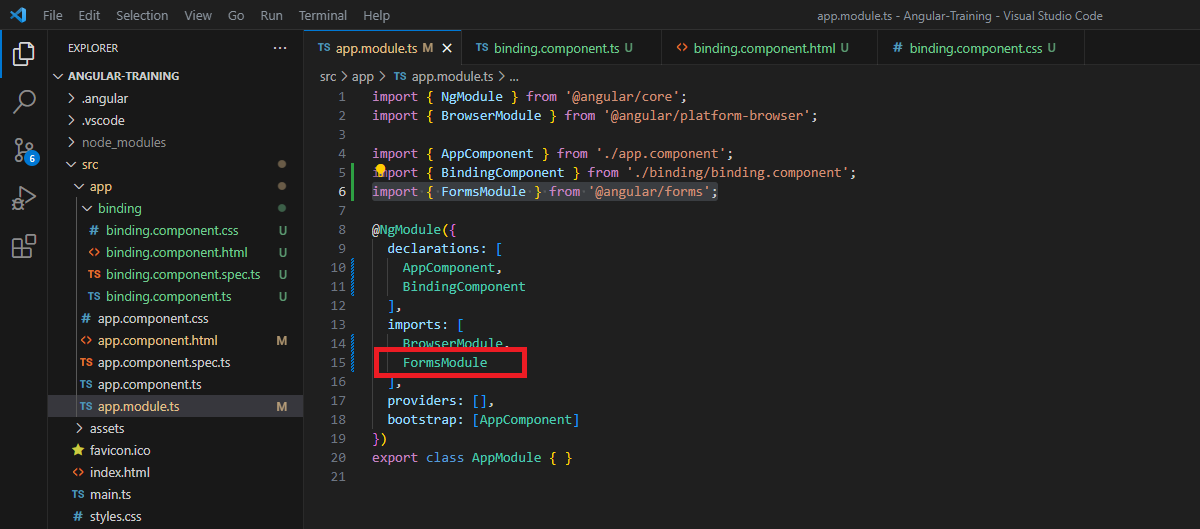
انگولار با استفاده از اتصال دوطرفه نه تنها تغییرهای متغیرهای ما را رصد می‌کند، بلکه اعمالی که از سوی کاربران انجام می‌گیرد را نیز ردگیری کرده و متغیرهای اتصال‌یافته را بر این اساس به‌روزرسانی می‌کند.



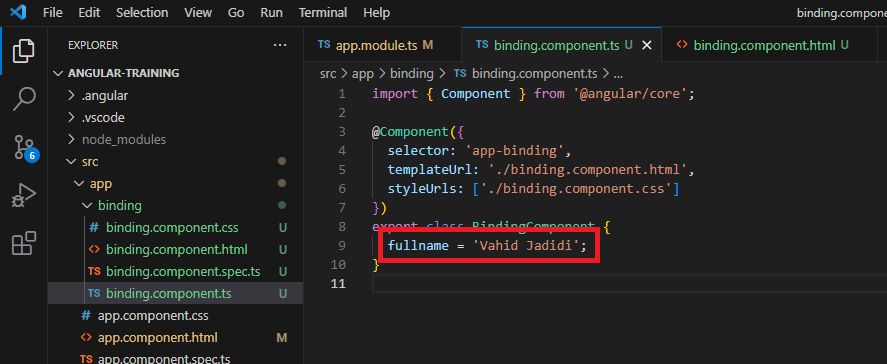
برای استفاده از این قابلیت کد زیر را در فایل app.module.ts اضافه می کنیم تا از کتابخانه‌های انگولار قابلیت FormsModule را به کامپوننت‌ها اضافه کنیم.

|  |
| --- |
| import { FormsModule } from '@angular/forms'; |

و پس از آن FormsModule را به imports ماژول اضافه می کنیم.



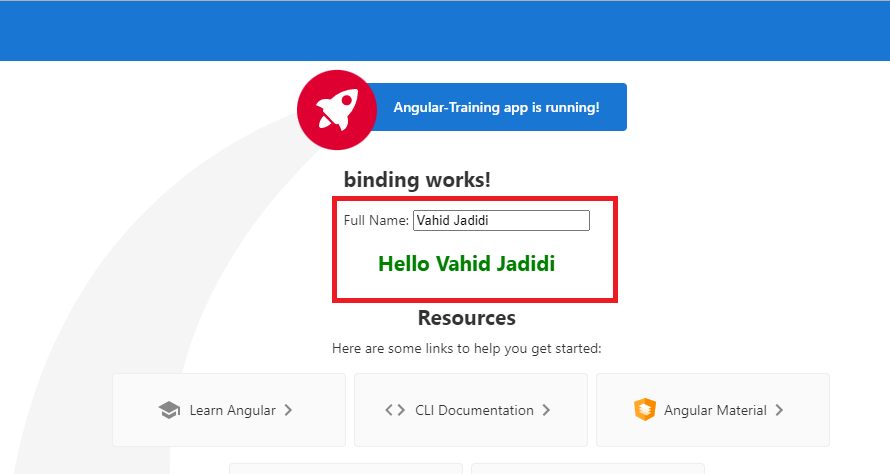
سپس یک متغیر در کامپوننت خود ایجاد می‌کنیم و آن را مقدار دهی می کنیم(نام کامپوننت من binding است).



و در فایل binding.component.html کد زیر را قرار می‌دهیم.

|  |
| --- |
| <p>Full Name: <input type="text" [(ngModel)]='fullname'></p>  <h2 class="welcome">Hello {{fullname}}</h2> |

اکنون دو تگ input و h2 بین قسمت منطق(logic) برنامه و قسمت نمایشی(UI) اتصال دو طرفه ایجاد کرده‌اند.



فراموش نشود که selector کامپوننت در فایل app.component.html قرار گیرد.

|  |
| --- |
| <app-binding></app-binding> |

## ng-content

از این قابلیت می توان برای ارسال المنت های HTML از کامپوننت ریشه(app.component.html) به کامپوننت های دیگر استفاده کرد.

به عنوان مثال کد زیر را در کامپوننت ریشه قرار دهید.

|  |
| --- |
| <app-binding>  <h1>Hello ...</h1>  <p>Welcome to my website</p>  </app-binding> |

تگ <app-binding> مربوط به سلکتور کامپوننت binding.component.html است که در کامپوننت ریشه app.component.html قرار می گیرد.

کد زیر را در کامپوننت binding.component.html قرار دهید.

|  |
| --- |
| <ng-content></ng-content> |

تگ <ng-content> را در هر جایی از کد HTML خود قرار دهید المنت های ارسال شده از کامپوننت ریشه در آنجا نمایش داده خواهند شد.

شما همچنین می توانید المنت های ارسال شده را Select کنید به عنوان مثال فقط تگ های p نمایش داده شوند و یا فقط المنت‌هایی که کلاس Active را دارند نمایش داده شوند. در زیر مثال هایی در این خصوص آورده شده است.

|  |
| --- |
| <ng-content select="p"></ng-content>  <ng-content select=".Active"></ng-content> |

## اشتراک گذاری داده بین کامپوننت‌های Parent و Child

یک الگوی رایج در Angular به اشتراک گذاری داده ها بین یک مؤلفه والد و یک یا چند مؤلفه فرزند است. این الگو را می‌توانید با دکوراتورهای @Input و @Output پیاده سازی کنید.

### ارسال داده به کامپوننت Child

با استفاده از دکوراتور @Input می‌توان مقدار یک property را از کامپوننت پدر(Parent) به کامپوننت فرزند(Child) ارسال کرد.

ابتدا یک property در کامپوننت app.component.ts ایجاد کنید و آن را مقدار دهی کنید.

|  |
| --- |
| Sent = 'Sent Text ...'; |

سلکتور مربوط به کامپوننت فرزند را در کامپوننت app.component.html قرار دهید.

|  |
| --- |
| <app-io [inputtext]="Sent"></app-io> |

با استفاده از [inputtext] مقدار property را برای کامپوننت فرزند ارسال کنید.

کد زیر را در کامپوننت فرزند با پسوند ts قرار دهید(io.component.ts).

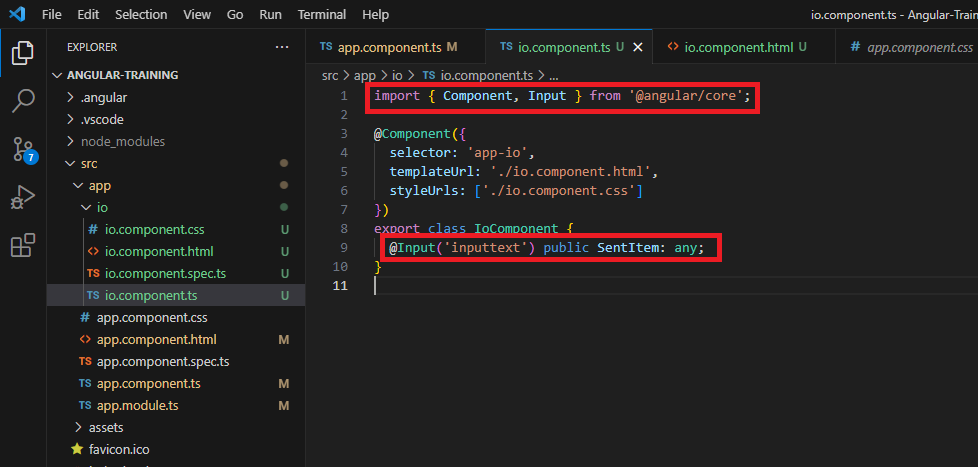
|  |
| --- |
| @Input('inputtext') public SentItem: any; |

مقدار inputtext همنام با مقدار وارد شده در سلکتور کامپوننت می باشد و SentItem یک نام اختیاری است.

و همچنین متد Input را به کامپوننت ایمپورت کنید.

|  |
| --- |
| import { Component, Input } from '@angular/core'; |

در شکل زیر نیز این کد ها نمایش داده شده اند.



در انتها کد زیر را برای نمایش داده ارسال شده به HTML کامپوننت فرزند اضافه کنید.

|  |
| --- |
| <p>io works! {{SentItem}}</p> |

### دریافت داده از کامپوننت Child