



red.es

PROFESIONALES
DIGITALES

Formación
Continua

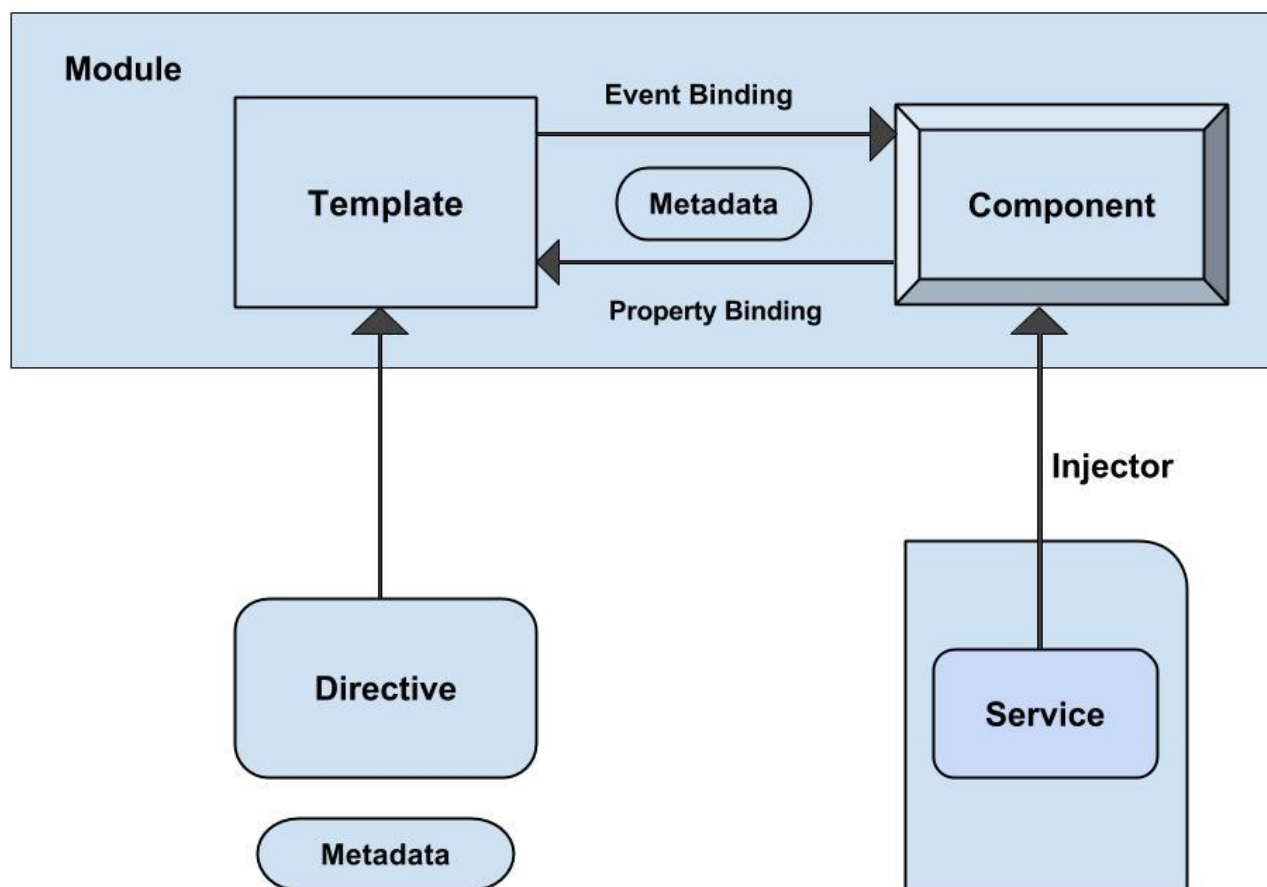
El FSE invierte en tu futuro



UNIÓN EUROPEA

Arquitectura de Angular

El FSE invierte en tu futuro





red.es

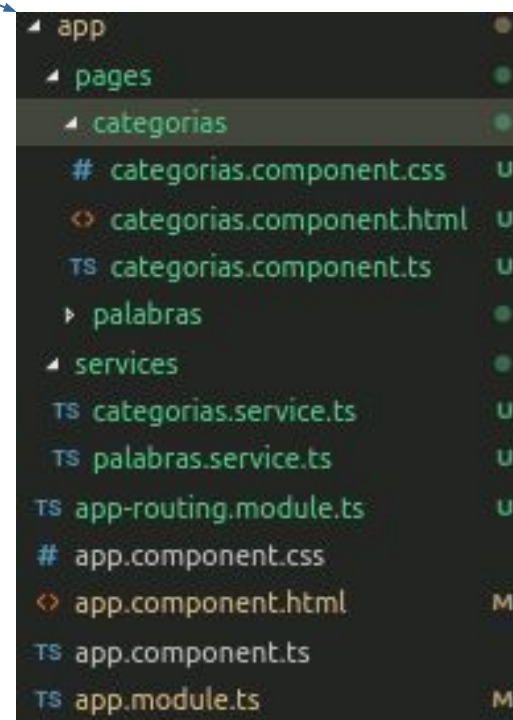
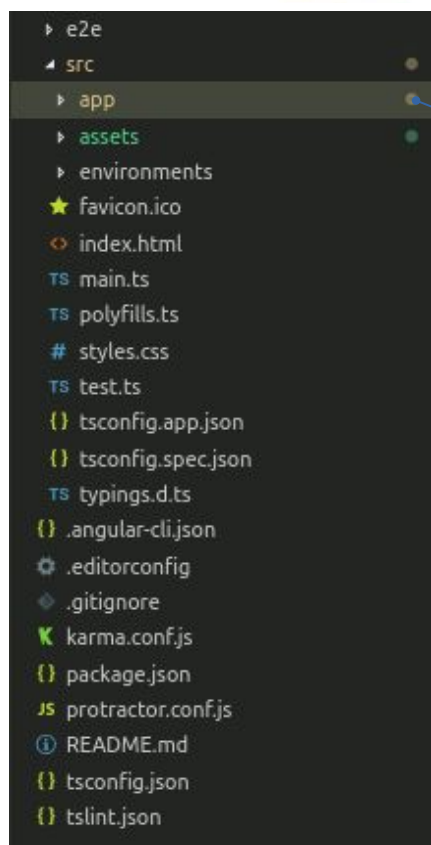
PROFESIONALES
DIGITALES

Formación
Continua



UNIÓN EUROPEA

El FSE invierte en tu futuro





red.es

PROFESIONALES
DIGITALES

Formación
Continua



UNIÓN EUROPEA

El FSE invierte en tu futuro

Módulos

- Las aplicaciones en Angular son modulares.
- Posee su propio sistema de modulación, NgModules
- Cada módulo puede contener componentes, servicios y otros archivos de código necesario.
- Al menos se debe contener un módulo por app.
- Se encargará de lanzar todos los archivos necesarios para nuestra aplicación.

src/app/app.module.ts

```
import { NgModule }      from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';

@NgModule({
  imports:    [ BrowserModule ],
  providers: [ Logger ],
  declarations: [ AppComponent ],
  exports:    [ AppComponent ],
  bootstrap: [ AppComponent ]
})
export class AppModule { }
```



red.es

PROFESIONALES
DIGITALES

Formación
Continua



UNIÓN EUROPEA

El FSE invierte en tu futuro

Componentes

- Definen la lógica de la aplicación.
- Angular crea, actualiza y destruye componentes a medida que el usuario se mueve a través de la aplicación.
- Cada vista de Angular esta compuesta por:
 - Un fichero HTML para la plantilla.
 - Un fichero TS para el componente.
 - Un fichero CSS para el diseño.

```
ng generate component name_component
```




red.es

PROFESIONALES
DIGITALES

Formación
Continua



UNIÓN EUROPEA

El FSE invierte en tu futuro

src/app/hero-list.component.ts (metadata)

```
@Component({
  selector: 'hero-list',
  templateUrl: './hero-list.component.html',
  providers: [ HeroService ]
})
export class HeroListComponent implements OnInit {
  /* . . . */
}
```

src/app/hero-list.component.ts (class)

```
export class HeroListComponent implements OnInit {
  heroes: Hero[];
  selectedHero: Hero;

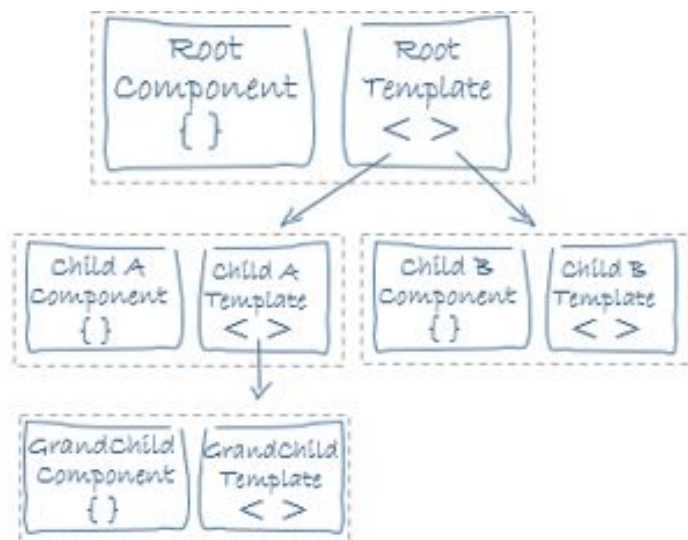
  constructor(private service: HeroService) { }

  ngOnInit() {
    this.heroes = this.service.getHeroes();
  }

  selectHero(hero: Hero) { this.selectedHero = hero; }
}
```

***EI FSE** invierte en tu futuro*

Plantillas y vistas



```
<h2>Hero List</h2>
```

```
<p><i>Pick a hero from the list</i></p>
```

```
<ul>
```

```
  <li *ngFor="let hero of heroes"
    (click)="selectHero(hero)">
    {{hero.name}}
```

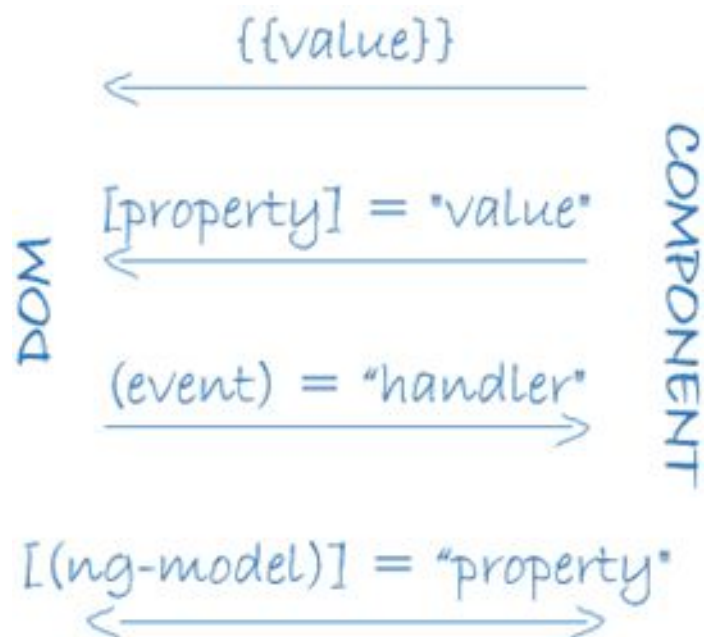
```
  </li>
```

```
</ul>
```

```
<app-hero-detail *ngIf="selectedHero"
[hero]="selectedHero"></app-hero-detail>
```

***EI FSE** invierte en tu futuro*

Data Binding



```
<li>{{hero.name}}</li>
```

```
<app-hero-detail  
[hero]="selectedHero"></app-hero-detail>
```

```
<li (click)="selectHero(hero)"></li>
```

```
<input [(ngModel)]="hero.name">
```




red.es

PROFESIONALES
DIGITALES

Formación
Continua



UNIÓN EUROPEA

El FSE invierte en tu futuro

Directivas

- Se usan para modificar el DOM y proporcionar a la plantilla de HTML un aspecto más dinámico

```
<li *ngFor="let hero of heroes"></li>
```

```
<app-hero-detail *ngIf="selectedHero"></app-hero-detail>
```

- `*ngFor`: muestra una etiqueta `` por cada objeto `hero` del array de `heroes`.
- `*ngIf`: incluye el componente solo si existe el objeto `selectedHero`.



red.es

PROFESIONALES
DIGITALES

Formación
Continua



UNIÓN EUROPEA

El FSE invierte en tu futuro

Servicios e ID

- Es una clase amplia que abarca cualquier valor, función o característica que necesita una app.
- Angular distingue los componentes de los servicios para mejorar la modularidad y la reutilización.
- Los services se encarga sobre todo de recuperar datos del servidor, validar la entrada de usuarios o iniciar sesión directamente.
 - Para implementar esta lógica en los componentes Angular utiliza la inyección de dependencias

```
ng generate service name_service
```



red.es

PROFESIONALES
DIGITALES

Formación
Continua



UNIÓN EUROPEA

El FSE invierte en tu futuro

```
@Injectable({})
```

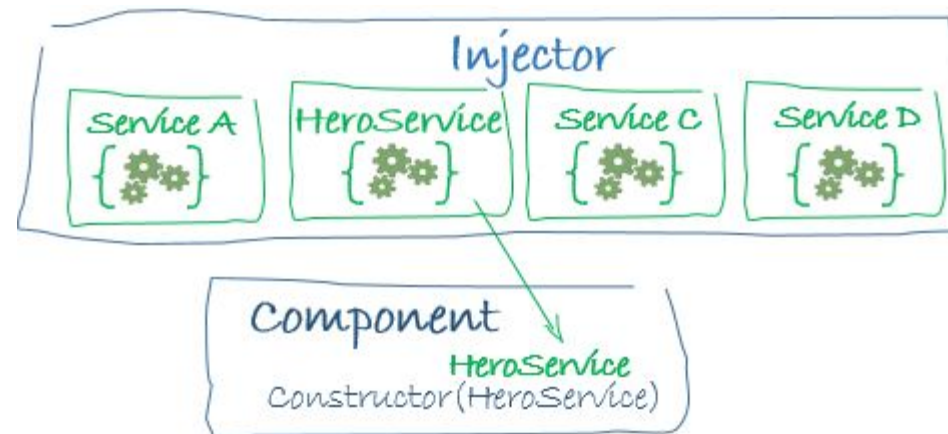
```
export class HeroService {  
  private heroes: Hero[] = [];
```

```
  constructor(  
    private backend: BackendService,  
    private logger: Logger) { }
```

```
  getHeroes() {  
    this.backend.getAll(Hero).subscribe( (heroes: Hero[]) => {  
      this.logger.log(`Fetched ${heroes.length} heroes.`);  
      this.heroes.push(...heroes); // fill cache  
    });  
    return this.heroes;  
  }  
}
```

El FSE invierte en tu futuro

- Enlaza los componentes con los servicios.
- Cuando Angular crea una instancia de una clase componente, determina qué servicios u otras dependencias necesita ese componente al observar los parámetros del constructor.
- Cuando un componente depende de un servicio, el inyector crea una instancia de ese servicio.





red.es

PROFESIONALES
DIGITALES

Formación
Continua



UNIÓN EUROPEA

El FSE invierte en tu futuro

Http Request

- Las aplicaciones de Angular se comunican con los servidores a través de los *services*.
- Para trabajar con las peticiones utilizaremos el objeto de *HttpClient* que se basa en la interfaz *XMLHttpRequest*.
- Un método de *HttpClient* no comienza su solicitud HTTP hasta que llamemos al método *.subscribe()*
- *HttpClient* admite solicitudes de GET, POST, PUT y DELETE.

```
import { HttpHeaders } from '@angular/common/http';

const httpOptions = {
  headers: new HttpHeaders({
    'Content-Type': 'application/json',
    'Authorization': 'my-auth-token'
  })
};
```




red.es

PROFESIONALES
DIGITALES

Formación
Continua

El FSE invierte en tu futuro



UNIÓN EUROPEA

Rutas para navegar entre componentes



red.es

PROFESIONALES
DIGITALES

Formación
Continua



UNIÓN EUROPEA

El FSE invierte en tu futuro

Routing

- El archivo de rutas indica a Angular que componentes cargar según la ruta que visitemos.
- Podremos pasar parámetros sin problema.
- El objeto encargado de realizar las rutas para Angular es *Router*.
- Pero para poder trabajar con este objeto y rutas en Angular debemos colocar esta etiqueta en el index.html

```
<base href="/">
```

- En el AppComponent debemos sustituir el código HTML por la siguiente etiqueta:

```
<router-outlet></router-outlet>
```



red.es

PROFESIONALES
DIGITALES

Formación
Continua



UNIÓN EUROPEA

El FSE invierte en tu futuro

```
ng generate module app-routing --flat --module=app
```

Angular te genera el siguiente código, pero no es el correcto a usar.

```
import { NgModule } from '@angular/core';
import { CommonModule } from '@angular/common';

@NgModule({
  imports: [
    CommonModule
  ],
  declarations: []
})

export class AppRoutingModule { }
```



red.es

PROFESIONALES
DIGITALES

Formación
Continua



UNIÓN EUROPEA

El FSE invierte en tu futuro

```
import { NgModule }      from '@angular/core';
import { RouterModule, Routes } from '@angular/router';

import { HeroesComponent } from './heroes/heroes.component';
import { HeroDetailComponent } from './heroDetail/heroDetail.component';
import { PageNotFoundComponent } from './page/page.component';

const routes: Routes = [
  { path: 'heroes', component: HeroesComponent },
  { path: 'heroes/:id', component: HeroDetailComponent },
  { path: '', redirectTo: '/heroes', pathMatch: 'full' },
  { path: '**', component: PageNotFoundComponent }
];

@NgModule({
  exports: [ RouterModule ],
  imports: [ RouterModule.forRoot(routes) ]
})
export class AppRoutingModule {}
```



red.es

PROFESIONALES
DIGITALES

Formación
Continua



UNIÓN EUROPEA

El FSE invierte en tu futuro

- Angular tiene dos maneras diferentes de navegar entre sus vistas:
 - Con el atributo `routerLink` en el HTML.

```
<a routerLink="/categorias">Categorias</a>  
<a routerLink="/palabras">Palabras</a>
```

- El objeto *router* desde el componente.



red.es

PROFESIONALES
DIGITALES

Formación
Continua



UNIÓN EUROPEA

El FSE invierte en tu futuro

- Extraer parámetros de la url

```
{ path: 'heroes/:id', component: HeroDetailComponent }
```

```
import { ActivatedRoute } from '@angular/router';  
  
...  
  
constructor(..., private activatedRoute: ActivatedRoute ) {  
  activatedRoute.params.subscribe( params => {  
    this.id = +params['id'];  
  });  
  ...  
}  
...
```