

A short introduction to





Angular

- frontend framework for developing web applications
- based on typescript
- open source
- development lead by google



Key terms

- Component (class that controls a view)
- Service (offers methods that can be used by components)
- Module (a package of certain components/services)
- Router (directs between pages)



Component

- most basic UI building block of an Angular app
- simplified: block of html + class instance
- used by calling its html-selector
- can include other components
- data binding (View is automatically updated when controller data changes)



Html file

src/app/hero-list.component.html

```
<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>
```



Class file

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; }  
}
```





Service

- provides methods which can be used by components
- for recurring tasks that should be shared between components
- also used so that components only perform tasks related to data presentation
- In order to be used in a component, the service has to be injected as a constructor argument of that component



Service

/src/app/message.service.ts

```
1. import { Injectable } from '@angular/core';
2.
3. @Injectable({
4.   providedIn: 'root',
5. })
6. export class MessageService {
7.   messages: string[] = [];
8.
9.   add(message: string) {
10.     this.messages.push(message);
11.   }
12.
13.   clear() {
14.     this.messages = [];
15.   }
16. }
```





Module

- a package of components/services
- used to group components/services that belong to a specific feature
- modules can also include other modules to gain access to their members
- the highest module is the root module



Module

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 { }
```



Router

- Each page has an URL
- Router navigates between pages
- Components are loaded based on the selected route



Router

app.routing.ts excerpt

```
const appRoutes: Routes = [  
  { path: 'crisis-center', component: CrisisListComponent },  
  { path: 'hero/:id',      component: HeroDetailComponent },  
  {  
    path: 'heroes',  
    component: HeroListComponent,  
    data: { title: 'Heroes List' }  
  },  
  { path: '',  
    redirectTo: '/heroes',  
    pathMatch: 'full'  
  },  
  { path: '**', component: PageNotFoundComponent }  
];
```





Sources

- Good documentation at www.angular.io/docs
- <https://medium.com/front-end-weekly/i-made-an-overview-presentation-of-angular-2-for-my-company-here-is-what-i-came-out-with-e180ea290a0f>