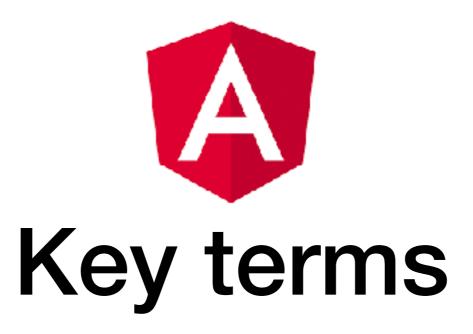
A short introduction to





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



- 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)



- 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)



src/app/hero-list.component.html



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; }
}
```



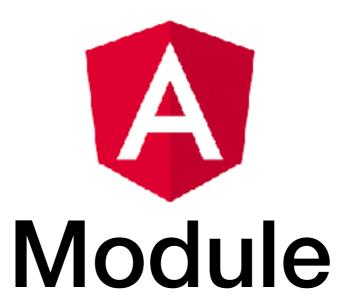
- provides methods which can be used by components
- for recurring tasks that should be shared between components
- also used so that components only perform tasks related to data presentation
- In order to be used in a component, the service has to be injected as a constructor argument of that component



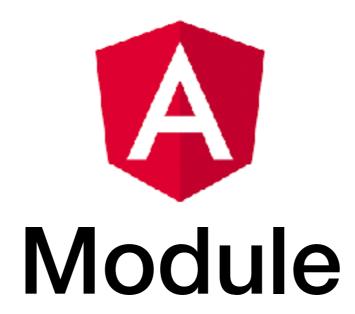
/src/app/message.service.ts import { Injectable } from '@angular/core'; 2. @Injectable({ 4. providedIn: 'root', 5. }) 6. export class MessageService { 7. messages: string[] = []; 8. add(message: string) { 9. this.messages.push(message); 10. 11. } 12. 13. clear() { this.messages = []; 14.

15. }

16. }



- 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



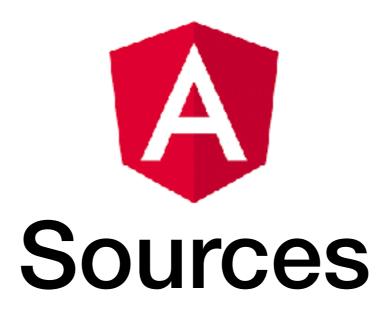


- Each page has an URL
- Router navigates between pages
- Components are loaded based on the selected route



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 }
];
```



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