**Section06 Services and DI**

**Notes:-**

**1-services: its acts as central business repository which you can store data**

Application

Log Service

App Component

User Service

About Component

User Component

User Details Component

**//on the app.component**

**<div class="container">**

**<div class="row">**

**<div class="col-xs-12 col-md-8 col-md-offset-2">**

**<app-new-account (accountAdded)="onAccountAdded($event)"></app-new-account>**

**<hr>**

**<app-account \*ngFor="let acc of accounts; let i = index" [account]="acc" [id]="i"**

**(statusChanged)="onStatusChanged($event)"></app-account>**

**</div></div></div>**

**import { Component } from '@angular/core';**

**import { AccountModel } from './models/accountmodel';**

**@Component({**

**selector: 'app-root',**

**templateUrl: './app.component.html',**

**styleUrls: ['./app.component.css']})**

**export class AppComponent {**

**accounts:AccountModel[] = [**

**new AccountModel('Master Account','active'),**

**new AccountModel('Testaccount','inactive'),**

**new AccountModel('Hidden Account','unknown')];**

**onAccountAdded(newAccount: {name: string, status: string}) {this.accounts.push(newAccount);}**

**onStatusChanged(updateInfo: {id: number, newStatus: string}) {**

**this.accounts[updateInfo.id].status = updateInfo.newStatus;}}**

**//on the account.component we set the following code**

**<div class="row">**

**<div class="col-xs-12 col-md-8 col-md-offset-2">**

**<h5>{{ account.name }}</h5>**

**<hr>**

**<p>This account is {{ account.status }}</p>**

**<button class="btn btn-default" (click)="onSetTo('active')">Set to 'active'</button>**

**<button class="btn btn-default" (click)="onSetTo('inactive')">Set to 'inactive'</button>**

**<button class="btn btn-default" (click)="onSetTo('unknown')">Set to 'unknown'</button>**

**</div></div>**

**import { Component, EventEmitter, Input, Output } from '@angular/core';**

**import { AccountModel } from 'src/app/models/accountmodel';**

**@Component({**

**selector: 'app-account',**

**templateUrl: './account.component.html',**

**styleUrls: ['./account.component.css']})**

**export class AccountComponent {**

**@Input() account: AccountModel;**

**@Input() id: number;**

**@Output() statusChanged = new EventEmitter<{id: number, newStatus: string}>();**

**onSetTo(status: string) {**

**this.statusChanged.emit({id: this.id, newStatus: status});**

**console.log('A server status changed, new status: ' + status);}}**

**//on the new-account.component**

**<div class="row">**

**<div class="col-xs-12 col-md-8 col-md-offset-2">**

**<div class="form-group">**

**<label>Account Name</label>**

**<input type="text" class="form-control" #accountName>**

**</div>**

**<div class="form-group">**

**<select class="form-control" #status>**

**<option value="active">Active</option>**

**<option value="inactive">Inactive</option>**

**<option value="hidden">Hidden</option>**

**</select>**

**</div>**

**<button class="btn btn-primary" (click)="onCreateAccount(accountName.value, status.value)">**

**Add Account**

**</button>**

**</div></div>**

**import { Component, EventEmitter, Output } from '@angular/core';**

**@Component({**

**selector: 'app-new-account',**

**templateUrl: './new-account.component.html',**

**styleUrls: ['./new-account.component.css']})**

**export class NewAccountComponent {**

**@Output() accountAdded = new EventEmitter<{name: string, status: string}>();**

**onCreateAccount(accountName: string, accountStatus: string) {**

**this.accountAdded.emit({**

**name: accountName,**

**status: accountStatus});**

**console.log('A server status changed, new status: ' + accountStatus);}}**

**Lesson02 What is the Dependency Injection?**

**Notes:-**

**1-Dependency Injection: It will inject of the instance of class into the component**

**2-we can provide dependency injection on the component level or on the module level**

**@Component({**

**selector: 'app-new-account',**

**templateUrl: './new-account.component.html',**

**styleUrls: ['./new-account.component.css'],**

**providers:[LoggingService]**})

**constructor(public loggingService: LoggingService) {}**

**onCreateAccount(accountName: string, accountStatus: string) {**

**this.accountAdded.emit({name: accountName,status: accountStatus,});**

**this.loggingService.logStatus(accountStatus);}**

**Lesson03 Creating Data Service**

**Notes:-**

**1-we make data store which is accounts is reference type which reflect the addition or update on the account list as below**

**import { Injectable } from '@angular/core';**

**import { AccountModel } from '../models/accountmodel';**

**@Injectable({providedIn: 'root'})**

**export class AccountsService {**

**constructor() { }**

**//array is reference type**

**accounts:AccountModel[] = [**

**new AccountModel('Master Account','active'),**

**new AccountModel('Testaccount','inactive'),**

**new AccountModel('Hidden Account','unknown')];**

**onAccountAdded(newAccount: {name: string, status: string}) {this.accounts.push(newAccount);}**

**onStatusChanged(id:number,newStatus:string) {this.accounts[id].status = newStatus;}}**

**import { Component, OnInit } from '@angular/core';**

**import { AccountModel } from './models/accountmodel';**

**import { AccountsService } from './services/accounts.service';**

**@Component({**

**selector: 'app-root',**

**templateUrl: './app.component.html',**

**styleUrls: ['./app.component.css']})**

**export class AppComponent implements OnInit {**

**account:AccountModel[] = [];**

**constructor(public accountService:AccountsService){}**

**ngOnInit(): void {this.account = this.accountService.accounts}}**

**import { Component, EventEmitter, Output } from '@angular/core';**

**import { AccountModel } from 'src/app/models/accountmodel';**

**import { AccountsService } from 'src/app/services/accounts.service';**

**import { LoggingService } from 'src/app/services/logging.service';**

**@Component({**

**selector: 'app-new-account',**

**templateUrl: './new-account.component.html',**

**styleUrls: ['./new-account.component.css']})**

**export class NewAccountComponent {**

**@Output() accountAdded = new EventEmitter<{ name: string; status: string }>();**

**constructor(public loggingService: LoggingService,private accountService:AccountsService) {}**

**onCreateAccount(accountName: string, accountStatus: string) {**

**this.accountService.onAccountAdded(new AccountModel(accountName,accountStatus));}}**

**import { Component, EventEmitter, Input, Output } from '@angular/core';**

**import { AccountModel } from 'src/app/models/accountmodel';**

**import { AccountsService } from 'src/app/services/accounts.service';**

**@Component({**

**selector: 'app-account',**

**templateUrl: './account.component.html',**

**styleUrls: ['./account.component.css']})**

**export class AccountComponent {**

**@Input() account: AccountModel;**

**@Input() id: number;**

**constructor(public accountService:AccountsService){}**

**onSetTo(status: string) {**

**this.accountService.onStatusChanged(this.id,status);}}**

**Lesson04 understanding the hierarchical injector**

**Notes:-**

**1-the angular service acts as hierarchical injector which means all the component that inject the service and their child will receive the same instance of the service**

**App Module: same instance of service is available application wide**

**App Component: same instance of service is available for all components but not for the other services**

**Any other component: same instance of service is available for the component and all its child components**

**Example:-**

**//on the app.component**

**<div class="container">**

**<div class="row">**

**<div class="col-xs-12 col-md-8 col-md-offset-2">**

**<app-new-account></app-new-account>**

**<hr>**

**<app-account \*ngFor="let acc of accounts; let i = index" [account]="acc" [id]="i"></app-account>**

**</div></div></div>**

**import { Component, OnInit } from '@angular/core';**

**import { AccountModel } from './models/accountmodel';**

**import { AccountsService } from './services/accounts.service';**

**@Component({**

**selector: 'app-root',**

**templateUrl: './app.component.html',**

**styleUrls: ['./app.component.css']})**

**export class AppComponent implements OnInit {**

**accounts:AccountModel[] = [];**

**constructor(public accountService:AccountsService){}**

**ngOnInit(): void {this.accounts = this.accountService.accounts}}**

**//on the account.component**

**<div class="row">**

**<div class="col-xs-12 col-md-8 col-md-offset-2">**

**<h5>{{ account.name }}</h5>**

**<hr>**

**<p>This account is {{ account.status }}</p>**

**<button class="btn btn-default" (click)="onSetTo('active')">Set to 'active'</button>**

**<button class="btn btn-default" (click)="onSetTo('inactive')">Set to 'inactive'</button>**

**<button class="btn btn-default" (click)="onSetTo('unknown')">Set to 'unknown'</button>**

**</div></div**

**import { Component, EventEmitter, Input, Output } from '@angular/core';**

**import { AccountModel } from 'src/app/models/accountmodel';**

**import { AccountsService } from 'src/app/services/accounts.service';**

**@Component({**

**selector: 'app-account',**

**templateUrl: './account.component.html',**

**styleUrls: ['./account.component.css']})**

**export class AccountComponent {**

**@Input() account: AccountModel;**

**@Input() id: number;**

**constructor(public accountService:AccountsService){}**

**onSetTo(status: string) {this.accountService.onStatusChanged(this.id,status);}}**

**//on the new-account.component**

**<div class="row">**

**<div class="col-xs-12 col-md-8 col-md-offset-2">**

**<div class="form-group">**

**<label>Account Name</label>**

**<input type="text" class="form-control" #accountName>**

**</div>**

**<div class="form-group">**

**<select class="form-control" #status>**

**<option value="active">Active</option>**

**<option value="inactive">Inactive</option>**

**<option value="hidden">Hidden</option>**

**</select>**

**</div>**

**<button class="btn btn-primary" (click)="onCreateAccount(accountName.value, status.value)">**

**Add Account</button></div></div>**

**import { Component, EventEmitter, Output } from '@angular/core';**

**import { AccountModel } from 'src/app/models/accountmodel';**

**import { AccountsService } from 'src/app/services/accounts.service';**

**import { LoggingService } from 'src/app/services/logging.service';**

**@Component({**

**selector: 'app-new-account',**

**templateUrl: './new-account.component.html',**

**styleUrls: ['./new-account.component.css']})**

**export class NewAccountComponent {**

**constructor(public loggingService: LoggingService,private accountService:AccountsService) {}**

**onCreateAccount(accountName: string, accountStatus: string) {**

**this.accountService.onAccountAdded(new AccountModel(accountName,accountStatus));}}**

**Lesson05 Injecting service into service**

**Notes:-**

**1-you can inject service inside another service and using in component so technically you don’t need to create new instance of the inner service**

**import { Injectable } from '@angular/core';**

**import { AccountModel } from '../models/accountmodel';**

**import { LoggingService } from './logging.service';**

**@Injectable({providedIn: 'root'})**

**export class AccountsService {**

**constructor(public loggingService:LoggingService) { }**

**//array is reference type**

**accounts:AccountModel[] = [**

**new AccountModel('Master Account','active'),**

**new AccountModel('Testaccount','inactive'),**

**new AccountModel('Hidden Account','unknown')];**

**onAccountAdded(newAccount: {name: string, status: string}) {**

**this.accounts.push(newAccount);**

**this.loggingService.logStatus(newAccount.status);}**

**onStatusChanged(id:number,newStatus:string) {**

**this.accounts[id].status = newStatus;**

**this.loggingService.logStatus(newStatus);}}**

**import { Injectable } from '@angular/core';**

**import { AccountsService } from './accounts.service';**

**//with the decorator we know this is service such as the component declared with @Component**

**@Injectable({providedIn: 'root'})**

**export class LoggingService {**

**logStatus(status:string){console.log('A server status changed, new status: ' + status);}}**

**Lesson06 using services for cross-component communications**

**Notes:-**

**1-without services we have to using @output and @Input to emit event and receive value**

**2-we can make communication between components as channel through using Event Emitter on the service level because event emitter is kind of observable channel**

**3-with new syntax of**

**@Injectable({providedIn: 'root'})**

**export class MyService { ... }**

**import { MyService } from './path/to/my.service';**

**@NgModule({providers: [MyService]})**

**Export class AppModule { ... }** **does offer one advantage though: Services can be loaded lazily by Angular (behind the scenes) and redundant code can be removed automatically**

**Example:-**

**import { EventEmitter, Injectable } from '@angular/core';**

**import { AccountModel } from '../models/accountmodel';**

**import { LoggingService } from './logging.service';**

**@Injectable({providedIn: 'root'})**

**export class AccountsService {**

**constructor(public loggingService:LoggingService) { }**

**//we using event emitter across multiple component as publish subscribe pattern**

**statusUpdated = new EventEmitter<string>();**

**//array is reference type**

**accounts:AccountModel[] = [**

**new AccountModel('Master Account','active'),**

**new AccountModel('Testaccount','inactive'),**

**new AccountModel('Hidden Account','unknown')];**

**onAccountAdded(newAccount: {name: string, status: string}) {**

**this.accounts.push(newAccount);**

**this.loggingService.logStatus(newAccount.status);}**

**onStatusChanged(id:number,newStatus:string) {**

**this.accounts[id].status = newStatus;**

**this.loggingService.logStatus(newStatus);}}**

**//on new account component we emit the event emitter**

**import { Component, EventEmitter, Output } from '@angular/core';**

**import { AccountModel } from 'src/app/models/accountmodel';**

**import { AccountsService } from 'src/app/services/accounts.service';**

**import { LoggingService } from 'src/app/services/logging.service';**

**@Component({**

**selector: 'app-new-account',**

**templateUrl: './new-account.component.html',**

**styleUrls: ['./new-account.component.css']})**

**export class NewAccountComponent {**

**constructor(public loggingService: LoggingService,private accountService:AccountsService) {}**

**onCreateAccount(accountName: string, accountStatus: string) {**

**this.accountService.onAccountAdded(new AccountModel(accountName,accountStatus));**

**this.accountService.statusUpdated.emit('status updated to '+accountStatus);}}**

**//on account component we emit the event emitter**

**import { Component, EventEmitter, Input, Output } from '@angular/core';**

**import { AccountModel } from 'src/app/models/accountmodel';**

**import { AccountsService } from 'src/app/services/accounts.service';**

**@Component({**

**selector: 'app-account',**

**templateUrl: './account.component.html',**

**styleUrls: ['./account.component.css']})**

**export class AccountComponent {**

**@Input() account: AccountModel;**

**@Input() id: number;**

**constructor(public accountService:AccountsService){}**

**onSetTo(status: string) {**

**this.accountService.onStatusChanged(this.id,status);**

**this.accountService.statusUpdated.emit('status created to '+status);}}**

**//on app component we subscribe the event emitter**

**import { Component, OnInit } from '@angular/core';**

**import { AccountModel } from './models/accountmodel';**

**import { AccountsService } from './services/accounts.service';**

**@Component({**

**selector: 'app-root',**

**templateUrl: './app.component.html',**

**styleUrls: ['./app.component.css']})**

**export class AppComponent implements OnInit {**

**accounts:AccountModel[] = [];**

**constructor(public accountService:AccountsService){**

**this.accountService.statusUpdated.subscribe((msg:string) => alert(msg));}**

**ngOnInit(): void {this.accounts = this.accountService.accounts}}**

**Assignment**

**Notes:-**

**\*we want to make service that used to store active and inactive users and switch between them using data storage on the service layer**

**//on the app.component**

**<div class="container">**

**<div class="row">**

**<div class="col-xs-12 col-md-8 col-md-offset-2">**

**<app-active-users></app-active-users>**

**<app-inactive-users></app-inactive-users>**

**</div></div></div>**

**//on the active.users.component**

**<h3>Active Users</h3>**

**<ul class="list-group">**

**<li class="list-group-item" \*ngFor="let user of users; let i = index">**

**{{ user }} | <a href="#" (click)="onSetToInactive(i)">Set to Inactive</a>**

**</li></ul>**

**import { Component, OnInit } from '@angular/core';**

**import { UsersService } from 'src/app/services/users.service';**

**@Component({**

**selector: 'app-active-users',**

**templateUrl: './active-users.component.html',**

**styleUrls: ['./active-users.component.css']})**

**export class ActiveUsersComponent implements OnInit {**

**users: string[];**

**constructor(private userService: UsersService) {}**

**ngOnInit() {this.users = this.userService.activeUsers;}**

**onSetToInactive(id: number) {**

**this.userService.setToInactive(id);}}**

**<h3>Inactive Users</h3>**

**<ul class="list-group">**

**<li class="list-group-item" \*ngFor="let user of users; let i = index">**

**{{ user }} | <a href="#" (click)="onSetToActive(i)">Set to Active</a>**

**</li></ul>**

**import { Component, OnInit } from '@angular/core';**

**import { UsersService } from 'src/app/services/users.service';**

**@Component({**

**selector: 'app-inactive-users',**

**templateUrl: './inactive-users.component.html',**

**styleUrls: ['./inactive-users.component.css']})**

**export class InactiveUsersComponent implements OnInit {**

**users: string[];**

**constructor(private userService: UsersService) {}**

**ngOnInit() {**

**this.users = this.userService.inactiveUsers;}**

**onSetToActive(id: number) {**

**this.userService.setToActive(id);}}**

**//we make CounterService injectable to the Userservice which is injectable with the components**

**import { Injectable } from '@angular/core';**

**@Injectable({providedIn: 'root'})**

**export class CounterService {**

**activeToInactiveCounter = 0;**

**inactiveToActiveCounter = 0;**

**incrementActiveToInactive() {**

**this.activeToInactiveCounter++;**

**console.log('Active to Inactive: ' + this.activeToInactiveCounter);}**

**incrementInActiveToActive() {**

**this.inactiveToActiveCounter++;**

**console.log('Inactive to Active: ' + this.inactiveToActiveCounter);}}**

**import { Injectable } from '@angular/core';**

**import { CounterService } from './counter.service';**

**@Injectable({providedIn: 'root'})**

**export class UsersService {**

**activeUsers = ['Max', 'Anna'];**

**inactiveUsers = ['Chris', 'Manu'];**

**constructor(private counterService: CounterService) {}**

**setToActive(id: number) {**

**this.activeUsers.push(this.inactiveUsers[id]);**

**this.inactiveUsers.splice(id, 1);**

**this.counterService.incrementInActiveToActive();}**

**setToInactive(id: number) {**

**this.inactiveUsers.push(this.activeUsers[id]);**

**this.activeUsers.splice(id, 1);**

**this.counterService.incrementActiveToInactive();}}**