

FIT2095 e-Business software technologies - S2 2021

[Dashboard](#) / [My units](#) / [FIT2095_S2_2021](#) / [Assessments](#) / [Week 10: Workshop Quiz](#)**Started on** Monday, 4 October 2021, 6:09 PM**State** Finished**Completed on** Tuesday, 5 October 2021, 1:25 PM**Time taken** 19 hours 16 mins**Grade** 10.00 out of 10.00 (100%)[Print friendly format](#)Question **1**

Complete

Mark 10.00 out of 10.00

Question 1

href is a attribute of HTML, it reload the page while click

routerLink is a attribute of angular, it can lead to other component without reloading page, because it loading all page in the first time

major difference:

href end the state of page and losing data(e.g like losing data of textbox) after navigation, but routerLink won't. Because routerLink reload the angular inside resource not reload the browser's page.

Comment:

Question **2**

Complete

Not graded

Question 2

The one similar is they all accept an input value and return a transformed value and structure look similar:

Service:

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

```
@Injectable()
```

```

@Injectable({
  providedIn: 'root',
})
export class Service {

```

```

  constructor() { }

```

```

}

```

Pipe:

```

import { Pipe, PipeTransform } from '@angular/core';

@Pipe({
  name: 'w10strsub'
})
export class W10strsubPipe implements PipeTransform {

  transform(value: string, ...args: number[]): string {
    return null;
  }

}

```

One difference between pipe and service is pipe used for transforming strings, currency amounts, dates and other data for display, but service can do more and it can contain more methods and perform multiple tasks.

Question 3

Complete

Not graded

Question 3

```

import { Component, OnInit } from "@angular/core";
import { DatabaseService } from "../database.service";
import { Router } from "@angular/router";

@Component({
  selector: "app-addactor",
  templateUrl: "../addactor.component.html",
  styleUrls: ["../addactor.component.css"],
})
export class AddactorComponent {
  fullName: string = "";
  bYear: number = 0;
  actorId: string = "";

  constructor(private dbService: DatabaseService, private router: Router) {}

  onSaveActor() {
    let obj = { name: this.fullName, bYear: this.bYear };
    this.dbService.createActor(obj).subscribe(result => {
      this.router.navigate(["/listactors"]);
    });
  }
}

```

In order to navigate from A to B, we need to use navigate method of router in Angular. In the example, in export class of add actor, we use navigate method

```

this.router.navigate(["/listactors"]);

```

inside of the subscribe method. Then the add actor component will navigate to list actor component while add success

Question 4

Complete

Not graded

Question 4

```
import { BrowserModule } from "@angular/platform-browser";
import { NgModule } from "@angular/core";
import { AppComponent } from "./app.component";
import { DatabaseService } from "./database.service";
import { HttpClientModule } from "@angular/common/http";
import { FormsModule } from "@angular/forms";
import { SignupComponent } from "./signup/signupComponent.component";
import { DashboardComponent } from "./dashboard/dashboardComponent.component";
import { InvalidComponent } from "./invalid/invalidComponent.component";

const appRoutes: Routes = [
  { path: "signup", component: SignupComponent },
  { path: "dashboard", component: DashboardComponent },
  { path: "invalid", component: InvalidComponent },
  { path: "", redirectTo: "/dashboard" pathMatch: "full" },
  { path: "**", redirectTo: "/invalid" },
];
@NgModule({
  declarations: [
    AppComponent,
    ListactorsComponent,
    AddactorComponent,
    DeleteactorComponent
  ],
  imports: [
    RouterModule.forRoot(appRoutes),
    BrowserModule,
    HttpClientModule,
    FormsModule,
  ],
  providers: [DatabaseService],
  bootstrap: [AppComponent],
})
export class AppModule {}
```

Question 5

Complete

Not graded

Question 5

Faster:

1, PWA does not rely so much code, it's lighter than native app. Comparing with native app, smaller code means less loading time, which looks more responsive

2, PWA unlike native app, it loads longer when first access them. However, they load fast because PWAs are built to cache data in the background as interact with a site, and only the component will be modified but not modified the whole application

offline:

while there are no connection, the PWA can use the cache of the last interaction. For example, the news PWA can show some news while it is offline. However, while online and refresh, the app can update to the newest news

Question **6**

Not answered

Not graded

Question 6

Question **7**

Not answered

Not graded

Question 7

Question **8**

Not answered

Not graded

Question 8

◀ Week 9: Workshop Quiz

Jump to...

Week 11: Workshop Quiz ▶