**Generating the Service File**

The robust Anguar CLI tool will allow us to quickly and easily generate a service file for our project.

Hop into your console within the project folder and run the following command:

$ ng generate service data

**Note:** You can use *ng g s data* as a shorthand syntax for this command.

Our new service file is named *data*.

**Working in the Service File**

Open up the file, located at: */src/app/data.service.ts*:

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

@Injectable()

export class DataService {

constructor() { }

}

It looks similar to a component file, but it uses the *@Injectable()* decorator, which means we can import it into other components and access its properties and methods.

A great way of sharing data between components is to use the Rxjs *BehaviorSubject* library.

Update the service file to look like this:

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

import { BehaviorSubject } from 'rxjs/BehaviorSubject';

@Injectable()

export class DataService {

private goals = new BehaviorSubject<any>(['The initial goal', 'Another silly life goal']);

goal = this.goals.asObservable();

constructor() { }

changeGoal(goal) {

this.goals.next(goal)

}

}

This allows us to set the initial goal through *goals* as a *BehaviorSubject*, and then define a *goal* property as an observable.

We also created a *changeGoal* method that we will call in order to update the *goals* property.

Save this file.

**Importing the Service**

Each time you generate a service, you need to add it to the providers array of the */src/app/app.module.ts* file like so:

// Other imports removed for brevity

import { DataService } from './data.service';

@NgModule({

...

providers: [DataService],

...

})

**Using the Service in our Components**

Open up our */src/app/home/home.component.ts* file and import the service and add it in the constructor via dependency injection:

// Other imports removed for brevity

import { DataService } from '../data.service';

// @Component Decorator..

export class HomeComponent implements OnInit {

goals = [];

constructor(private \_data: DataService) { }

Notice that I have updated the *goals* array to empty.

Next, update the following methods beneath the constructor:

ngOnInit() {

this.itemCount = this.goals.length;

this.\_data.goal.subscribe(res => this.goals = res);

this.\_data.changeGoal(this.goals);

}

addItem() {

this.goals.push(this.goalText);

this.goalText = '';

this.itemCount = this.goals.length;

this.\_data.changeGoal(this.goals);

}

removeItem(i) {

this.goals.splice(i, 1);

this.\_data.changeGoal(this.goals);

}

We've referenced *this.\_data.changeGoal()* when the app loads, and when we add an item and remove an item.

Next, open up *about.component.ts* and update the code:

// Other imports..

import { DataService } from '../data.service';

// @Component Decorator

export class AboutComponent implements OnInit {

goals: any;

constructor(private route: ActivatedRoute, private router: Router, private \_data: DataService) {

this.route.params.subscribe(res => console.log(res.id));

}

ngOnInit() {

this.\_data.goal.subscribe(res => this.goals = res);

}

sendMeHome() {

this.router.navigate(['']);

}

}

And finally, update *about.component.html*:

<p>

This is what I'm all about. <a href="" (click)="sendMeHome()"><strong>Take me back</strong></a>.

</p>

<ul>

<li \*ngFor="let goal of goals">

{{ goal }}

</li>

</ul>

Great! Try it out!

You will see that when you click back and forth between the components, the goals data is shared. You can also use the form to add new goals and also remove them by clicking them on the list, and the data is persisted across the components.

**Going Forward**

You've come quite a long way since [**learning Angular 5**](https://coursetro.com/courses/19/Learn-Angular-5-from-Scratch---Angular-5-Tutorial). In the next and final lesson, we're going to take a look at how to deploy our Angular 5 app.

**Share this post**

**[https://coursetro.com/images/share-fb.png](http://www.facebook.com/share.php?u=https://coursetro.com/posts/code/110/Creating-and-Using-Angular-5-Services&title=Creating%20and%20Using%20Angular%205%20Services)** **[https://coursetro.com/images/share-tw.png](http://twitter.com/share?url=https://coursetro.com/posts/code/110/Creating-and-Using-Angular-5-Services&text=Creating%20and%20Using%20Angular%205%20Services)**

Say something about this awesome post!