

## Provide an Angular Service on a route!



Deborah Kurata  
19.5K subscribers

Subscribe

316

316

Share

Download

Thanks

...

4,409 views Jan 22, 2025 #angular #angulsarsignals #angularstatemanagementwithsignals

Did you know that you can provide a service on a route? Instead of using `ProvidedIn "root"` on the service or a component's providers array we can define a service for a route.

In this video, we examine the syntax and discuss the scope and lifetime of a service when that service is provided on a route. Then we add child routes to see how to share that service.

NOTE 1/22/2005:

The video didn't discuss the "when".

First, in most cases, provide services in "root". That way they are accessible everywhere in the application.

When should you provide a service on a route?

1) If you need different "versions" of the service for different purposes. (Here is a detailed example of that: [/ angular-route-level-providers-a-comprehens...](#))

2) If you want to limit access to the service. Say you have a route specifically for administrators and they have some specialized services. By defining those services on the route accessible only to administrators, you minimize the possibility of those services accidentally being injected into any other part of the codebase.

3) Separation of responsibilities. A developer on the team needs to add a set of features/routes around StarWars films (separate from vehicles). To minimize changes, the first thought it to add the code for those features in the Vehicle Service. But the Vehicle service is only injectable on the vehicles route. So they instead must create a Films Service thus keeping the code a bit better organized. (A bit farfetched example maybe.)

Screenshot of StackBlitz showing the `app.routes.ts` file and the `vehicle-selection.component.html` template. The `app.routes.ts` file defines routes for 'home', 'vehicles', and a wildcard redirect. The `vehicle-selection.component.html` template displays a Star Wars vehicle image.

```
TS vehicle.service.ts x TS vehicle-selection.component.ts x vehicle-selection.component.html x TS app.routes.ts x
1 import { Routes } from '@angular/router';
2 import { HomeComponent } from './home/home.component';
3 import { VehicleSelectionComponent } from './vehicles/vehicle-selection/vehicle-selection.component';
4
5 export const routes: Routes = [
6   { path: 'home', component: HomeComponent },
7   { path: 'vehicles', component: VehicleSelectionComponent },
8   { path: '**', redirectTo: 'home', pathMatch: 'full' }
9 ];
```

Star Wars Vehicle Sales  
Home   Vehicle List

Welcome to Star Wars Vehicle Sales

Screenshot of StackBlitz showing the `vehicle.service.ts` file and the `vehicle-selection.component.html` template. The `vehicle.service.ts` file contains an `@Injectable` annotation with `providedIn: 'root'`. The `vehicle-selection.component.html` template displays a Star Wars vehicle image.

```
TS vehicle.service.ts x TS vehicle-selection.component.ts x vehicle-selection.component.html x TS app.routes.ts x
1 import { HttpClient } from '@angular/common/http';
2 import { Injectable, computed, effect, inject, linkedSignal, signal } from '@angular/core';
3 import { rxResource } from '@angular/core/rxjs-interop';
4 import { delay, map } from 'rxjs';
5
6 @Injectable({
7   providedIn: 'root'
8 })
9 export class VehicleService {
10   private vehicleUrl = 'https://swapi.py4e.com/api/vehicles';
11
12   // Injected services
13   private http = inject(HttpClient);
14
15   // Signals managed by the service
16   selectedVehicle = signal<Vehicle | undefined>(undefined);
17
18   // Reset the quantity when the vehicle changes
19   quantity = linkedSignal({
20     source: this.selectedVehicle,
21     computation: (v) => {
22       if (v) {
23         return 1;
24       }
25       return 0;
26     }
27   );
28 }
```

Star Wars Vehicle Sales  
Home   Vehicle List

Welcome to Star Wars Vehicle Sales

stackblitz.com/~edit/providing-services-route-deborahk-wip?file=src/app/vehicles/vehicle-selection/vehicle-selection.component.ts

Providing Services in routes (WIP)

```

1 import { Component, inject } from '@angular/core';
2 import { FormsModule } from '@angular/forms';
3 import { VehicleService } from '../vehicle.service';
4
5 @Component({
6   selector: 'app-vehicle-selection',
7   imports: [FormsModule],
8   // providers: [VehicleService],
9   templateUrl: './vehicle-selection.component.html',
10  styleUrls: ['./vehicle-selection.component.css']
11 })
12 export class VehicleSelectionComponent {
13   pageTitle = "Select a Vehicle";
14
15   // Injected services
16   private vehicleService = inject(VehicleService);
17
18   // Signals to support the template
19   vehicles = this.vehicleService.vehicles;
20   isLoading = this.vehicleService.isLoading;
21   selectedVehicle = this.vehicleService.selectedVehicle;
22   quantity = this.vehicleService.quantity;
23   total = this.vehicleService.total;
24   color = this.vehicleService.color;
25 }

```

Star Wars Vehicle Sales

Welcome to Star Wars Vehicle Sales

[YouTube Channel](#)

stackblitz.com/~edit/providing-services-route-deborahk-wip?file=src/app/app.routes.ts

Providing Services in routes (WIP)

```

1 import { Routes } from '@angular/router';
2 import { HomeComponent } from './home/home.component';
3 import { VehicleSelectionComponent } from './vehicles/vehicle-selection/vehicle-selection.component';
4
5 export const routes: Routes = [
6   { path: 'home', component: HomeComponent },
7   { path: 'vehicles', component: VehicleSelectionComponent },
8   { path: '**', redirectTo: 'home', pathMatch: 'full' }
9 ];

```

Star Wars Vehicle Sales

Welcome to Star Wars Vehicle Sales

stackblitz.com/~edit/providing-services-route-deborahk-wip?file=src/app/vehicles/vehicle.service.ts

Providing Services in routes (WIP)

```

1 import { HttpClient } from '@angular/common/http';
2 import { Injectable, computed, effect, inject, linkedSignal, signal } from '@angular/core';
3 import { rxResource } from '@angular/core/rxjs-interop';
4 import { delay, map } from 'rxjs';
5
6 @Injectable({
7   providedIn: 'root'
8 })
9 export class VehicleService {
10   private vehicleUrl = 'https://swapi.py4e.com/api/vehicles';
11
12   // Injected services
13   private http = inject(HttpClient);
14
15   // Signals managed by the service
16   selectedVehicle = signal<Vehicle | undefined>(undefined);
17
18   // Reset the quantity when the vehicle changes
19   quantity = linkedSignal({
20     source: this.selectedVehicle,
21     computation: (v) => {
22       if (v) {
23         return 1;
24       }
25       return 0;
26     }
27   );

```

Star Wars Vehicle Sales

Welcome to Star Wars Vehicle Sales

[YouTube Channel](#)

This service is currently provided in **root** instead of in a component

stackblitz.com/~edit/providing-services-route-deborahk-wip?file=src/app/vehicles/vehicle.service.ts

```

4 import { delay, map } from 'rxjs';
5
6 @Injectable({
7   providedIn: 'root'
8 })
9 export class VehicleService {
10   private vehicleUrl = '';
11
12   // Injected services
13   private http = inject(HttpClient);
14
15   // Signals managed by
16   selectedVehicle = signal<Vehicle | undefined>(undefined);
17
18   // Reset the quantity
19   quantity = linkedSignal({
20     source: this.selectedVehicle,
21     computation: (v) =>
22       if (v) {
23         return 1;
24       }
25       return 0;
26   });
27 }

```

stackblitz.com/~edit/providing-services-route-deborahk-wip?file=src/app/vehicles/vehicle.service.ts

```

1 import { HttpClient } from '@angular/common/http';
2 import { Injectable, computed, effect, inject, linkedSignal, signal } from '@angular/core';
3 import { rxResource } from '@angular/core/rxjs-interop';
4 import { delay, map } from 'rxjs';
5
6 @Injectable({
7   providedIn: 'root'
8 })
9 export class VehicleService {
10   private vehicleUrl = 'https://swapi.py4e.com/api/vehicles';
11
12   // Injected services
13   private http = inject(HttpClient);
14
15   // Signals managed by the service
16   selectedVehicle = signal<Vehicle | undefined>(undefined);
17
18   // Reset the quantity when the vehicle changes
19   quantity = linkedSignal({
20     source: this.selectedVehicle,
21     computation: (v) => {
22       if (v) {
23         return 1;
24       }
25       return 0;
26     }
27   });

```

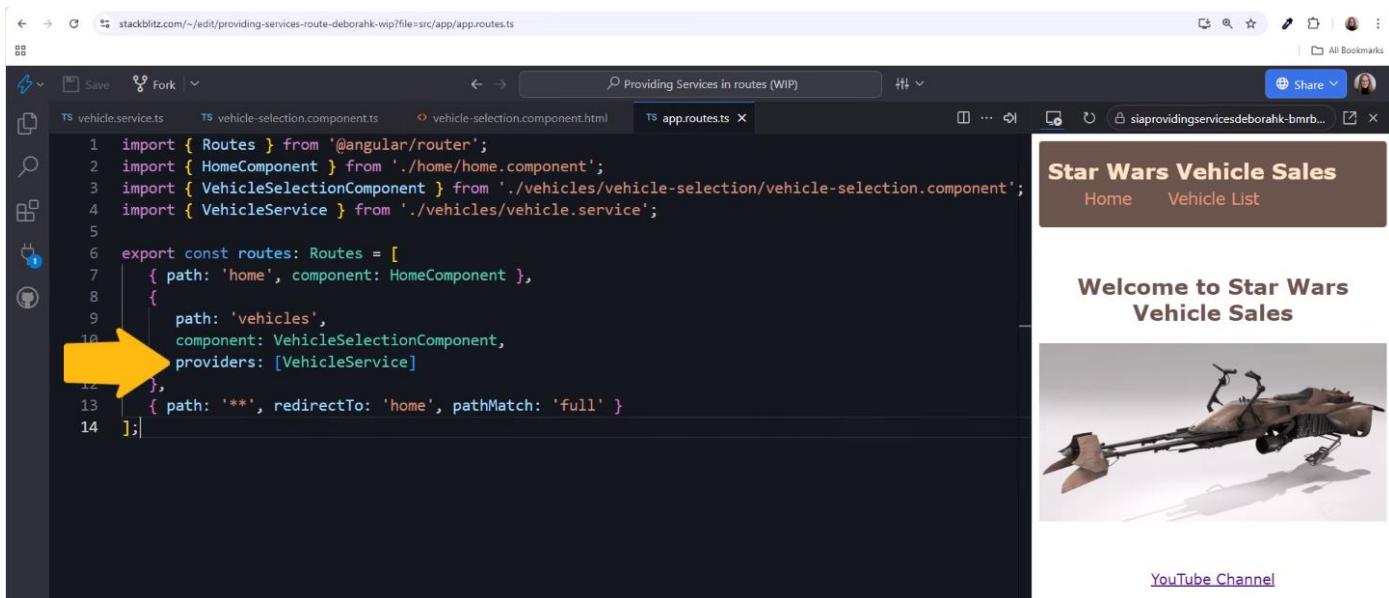
Instead of providing the **VehicleService** at root, let us now instead provide the **VehicleService** on the route that uses it

stackblitz.com/~edit/providing-services-route-deborahk-wip?file=src/app/app.routes.ts

```

1 import { Routes } from '@angular/router';
2 import { HomeComponent } from './home/home.component';
3 import { VehicleSelectionComponent } from './vehicles/vehicle-selection/vehicle-selection.component';
4
5 export const routes: Routes = [
6   { path: 'home', component: HomeComponent },
7   { path: 'vehicles', component: VehicleSelectionComponent },
8   { path: '**', redirectTo: 'home', pathMatch: 'full' }
9 ];

```



```

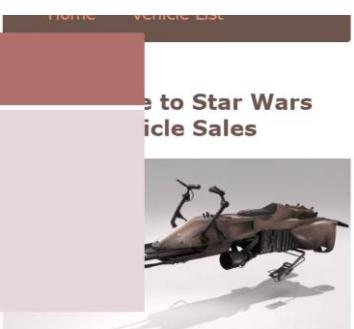
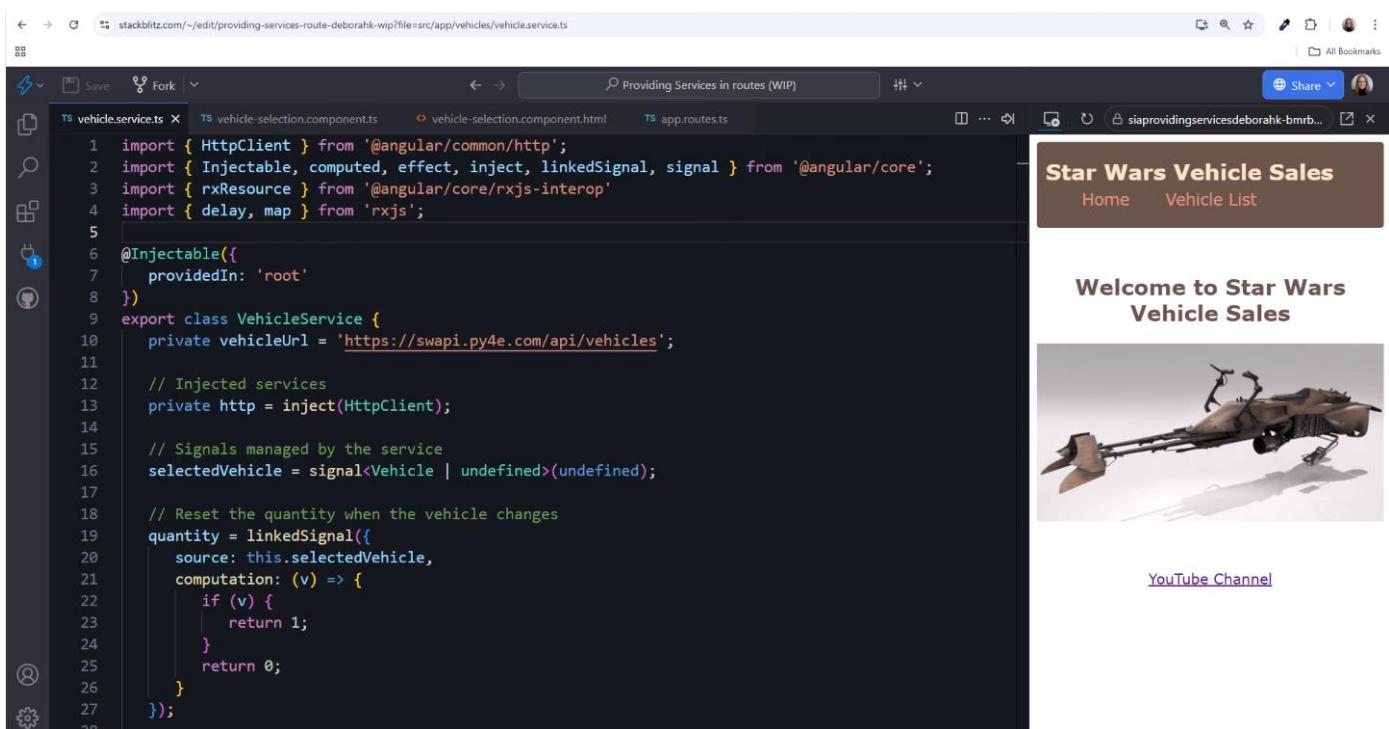
1 import { Routes } from '@angular/router';
2 import { HomeComponent } from './home/home.component';
3 import { VehicleSelectionComponent } from './vehicles/vehicle-selection/vehicle-selection.component';
4 import { VehicleService } from './vehicles/vehicle.service';
5
6 export const routes: Routes = [
7   { path: 'home', component: HomeComponent },
8   {
9     path: 'vehicles',
10    component: VehicleSelectionComponent,
11    providers: [VehicleService]
12  },
13  { path: '**', redirectTo: 'home', pathMatch: 'full' }
14 ];

```

The screenshot shows the StackBlitz editor with the file `app.routes.ts` open. A yellow arrow points to the line `providers: [VehicleService]`. The browser preview on the right shows the application's landing page with a Star Wars vehicle image.

This **VehicleService** can now only be injected in the specified component, **VehicleSelectionComponent**

Initialization	Scope	Lifetime
One instance shared with components on this route and child routes	State shared with components on the route or child routes	Life of the application

```

1 import { HttpClient } from '@angular/common/http';
2 import { Injectable, computed, effect, inject, linkedSignal, signal } from '@angular/core';
3 import { rxResource } from '@angular/core/rxjs-interop'
4 import { delay, map } from 'rxjs';
5
6 @Injectable({
7   providedIn: 'root'
8 })
9 export class VehicleService {
10   private vehicleUrl = 'https://swapi.py4e.com/api/vehicles';
11
12   // Injected services
13   private http = inject(HttpClient);
14
15   // Signals managed by the service
16   selectedVehicle = signal<Vehicle | undefined>(undefined);
17
18   // Reset the quantity when the vehicle changes
19   quantity = linkedSignal({
20     source: this.selectedVehicle,
21     computation: (v) => {
22       if (v) {
23         return 1;
24       }
25       return 0;
26     }
27   });
28 }

```

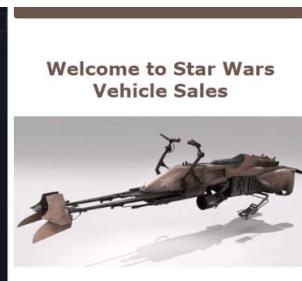
The screenshot shows the StackBlitz editor with the file `vehicle.service.ts` open. The browser preview on the right shows the application's landing page with a Star Wars vehicle image.

```

5   @Injectable({
6     providedIn: 'root'
7   })
8   export class VehicleService {
9     private vehicleUrl = 'https://swapi.py4e.com/api/vehicles';
10    // Injected services
11    private http = inject(HttpClient);
12    // Signals managed by the service
13    selectedVehicle = signal<Vehicle | undefined>(undefined);
14    // Reset the quantity when the vehicle changes
15    quantity = linkedSignal({
16      source: this.selectedVehicle,
17      computation: (v) => {
18        if (v) {
19          return 1;
20        }
21        return 0;
22      }
23    });
24    // Computed signals
25    total = computed(() => (this.selectedVehicle()?.cost_in_credits ?? 0) * this.quantity());
26  }
27}

```

Only provide a service using one technique



[YouTube Channel](#)

We should provide a service in **root** OR in a **component** OR in a **route** ONLY, not in multiple ways.

The screenshot shows the StackBlitz editor with the vehicle.service.ts file open. The code is identical to the one shown above, providing a service in the root module. To the right, the resulting application interface is displayed, featuring a title "Star Wars Vehicle Sales" and a "Vehicle List" button. Below the title is a "Welcome to Star Wars Vehicle Sales" message and an image of a speeder bike. A link to the "YouTube Channel" is present at the bottom.

The screenshot shows the StackBlitz editor with the vehicle-selection.component.ts file open. A yellow arrow points to the providers section of the component definition, which includes the VehicleService provider. To the right, the resulting application interface is displayed, showing the same title, "Vehicle List" button, and speeder bike image as the previous screenshot. A link to the "YouTube Channel" is present at the bottom.

stackblitz.com/~edit/providing-services-route-deborahk-wip?file=src/app/vehicles/vehicle-selection/vehicle-selection.component.ts

```

1 import { Component, inject } from '@angular/core';
2 import { FormsModule } from '@angular/forms';
3 import { VehicleService } from '../vehicle.service';
4
5 @Component({
6   selector: 'app-vehicle-selection',
7   imports: [FormsModule],
8   templateUrl: './vehicle-selection.component.html',
9   styleUrls: ['./vehicle-selection.component.css']
10})
11 export class VehicleSelectionComponent {
12   pageTitle = "Select a Vehicle";
13
14   // Injected services
15   private vehicleService = inject(VehicleService);
16
17   // Signals to support the template
18   vehicles = this.vehicleService.vehicles;
19   isLoading = this.vehicleService.isLoading;
20   selectedVehicle = this.vehicleService.selectedVehicle;
21   quantity = this.vehicleService.quantity;
22   total = this.vehicleService.total;
23   color = this.vehicleService.color;
24 }

```

Star Wars Vehicle Sales

Welcome to Star Wars Vehicle Sales



[YouTube Channel](#)

stackblitz.com/~edit/providing-services-route-deborahk-wip?file=src/app/app.routes.ts

```

1 import { Routes } from '@angular/router';
2 import { HomeComponent } from './home/home.component';
3 import { VehicleSelectionComponent } from './vehicles/vehicle-selection/vehicle-selection.component';
4 import { VehicleService } from './vehicles/vehicle.service';
5
6 export const routes: Routes = [
7   { path: 'home', component: HomeComponent },
8   {
9     path: 'vehicles',
10     component: VehicleSelectionComponent,
11     providers: [VehicleService]
12   },
13   { path: '**', redirectTo: 'home', pathMatch: 'full' }
14 ];

```

Star Wars Vehicle Sales

Welcome to Star Wars Vehicle Sales



We now only have the service provided in one place, the route

stackblitz.com/~edit/providing-services-route-deborahk-wip?file=src/app/vehicles/vehicle-selection/vehicle-selection.component.ts

```

1 import { Component, inject } from '@angular/core';
2 import { FormsModule } from '@angular/forms';
3 import { VehicleService } from '../vehicle.service';
4
5 @Component({
6   selector: 'app-vehicle-selection',
7   imports: [FormsModule],
8   templateUrl: './vehicle-selection.component.html',
9   styleUrls: ['./vehicle-selection.component.css']
10})
11 export class VehicleSelectionComponent {
12   pageTitle = "Select a Vehicle";
13
14   // Injected services
15   private vehicleService = inject(VehicleService);
16
17   // Signals to support the template
18   vehicles = this.vehicleService.vehicles;
19   isLoading = this.vehicleService.isLoading;
20   selectedVehicle = this.vehicleService.selectedVehicle;
21   quantity = this.vehicleService.quantity;
22   total = this.vehicleService.total;
23   color = this.vehicleService.color;
24 }

```

Star Wars Vehicle Sales

Welcome to Star Wars Vehicle Sales



[YouTube Channel](#)

stackblitz.com/~edit/providing-services-route-deborahk-wip?file=src/app/vehicles/vehicle.service.ts

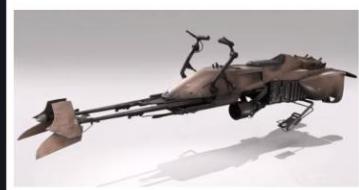
```

1 import { HttpClient } from '@angular/common/http';
2 import { Injectable, computed, effect, inject, linkedSignal, signal } from '@angular/core';
3 import { rxResource } from '@angular/core/rxjs-interop'
4 import { delay, map } from 'rxjs';
5
6 @Injectable()
7 export class VehicleService {
8   private vehicleUrl = 'https://swapi.py4e.com/api/vehicles';
9
10 // Injected services
11 private http = inject(HttpClient);
12
13 // Signals managed by the service
14 selectedVehicle = signal<Vehicle | undefined>(undefined);
15
16 // Reset the quantity when the vehicle changes
17 quantity = linkedSignal({
18   source: this.selectedVehicle,
19   computation: (v) => {
20     if (v) {
21       return 1;
22     }
23     return 0;
24   }
25 });
26
27 // Computed signals
28 total = computed(() => (this.selectedVehicle()?.cost_in_credits ?? 0) * this.quantity());

```

**Star Wars Vehicle Sales**

Welcome to Star Wars Vehicle Sales



[YouTube Channel](#)

stackblitz.com/~edit/providing-services-route-deborahk-wip?file=src/app/vehicles/vehicle.service.ts

```

30
31   private vehiclesResource = rxResource({
32     loader: () => this.http.get<VehicleResponse>(this.vehicleUrl).pipe(
33       map(vr => vr.results),
34       delay(2000)
35     )
36   );
37   vehicles = computed(() => this.vehiclesResource.value() ?? [] as Vehicle[]);
38   isLoading = this.vehiclesResource.isLoading;
39
40   vEff = effect(() => console.log('Selection:', this.selectedVehicle()));
41   vehiclesEff = effect(() => console.log('Vehicle data:', this.vehicles()));
42 }
43
44 export interface VehicleResponse {
45   count: number;
46   next: string;
47   previous: string;
48   results: Vehicle[];
49 }
50
51 export interface Vehicle {
52   name: string;
53   cost_in_credits: number;
54 }

```

**Star Wars Vehicle Sales**

Welcome to Star Wars Vehicle Sales



[YouTube Channel](#)

We use an effect to denote when a signal is initialized and when the value changes

When we reload the app, our service message does not appear because the service instance is not initialized yet.

Our service is now initialized and injected

stackblitz.com/~edit/providing-services-route-deborahk-wip?file=src/app/vehicles/vehicle.service.ts

```

30
31  private vehiclesResource = rxResource({
32    loader: () => this.http.get<VehicleResponse>(this.vehicleUrl).pipe(
33      map(vr => vr.results),
34      delay(2000)
35    );
36  });
37  vehicles = computed(() => this.vehiclesResource.value() ?? [] as Vehicle[]);
38  isLoading = this.vehiclesResource.isLoading;
39
40  vEff = effect(() => console.log('Selection:', this.selectedVehicle()));
41  vehiclesEff = effect(() => console.log('Vehicle data:', this.vehicles()));
42 }
43
44 export interface VehicleResponse {
45   count: number;
46   next: string;
47   previous: string;
48   results: Vehicle[];

```

X-34 landspeeder

Quantity: 1

Vehicle: X-34 landspeeder

Price: 10550

Total: 10550

Ln 5, Col 1 Spaces: 3 UTF-8 LF typeScript Layout: US

Elements Console Performance insights Recorder Sources Network Performance Memory Application Security Lighthouse Default levels 52 Issues: 2 4 46 2 hidden

vehicle.service.ts:40 vehicle.service.ts:41 vehicle.service.ts:41 vehicle.service.ts:40 vehicle.service.ts:40

Selection: undefined

Vehicle data: ▶ []

Vehicle data: ▶ (1) [{...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}]

selection: ▶ {name: 'X-34 Landspeeder', model: 'X-34 Landspeeder', manufacturer: 'SoroSuub Corporation', cost\_in\_credits: '10550', length: '3.4', ...}

stackblitz.com/~edit/providing-services-route-deborahk-wip?file=src/app/vehicles/vehicle.service.ts

```

30
31  private vehiclesResource = rxResource({
32    loader: () => this.http.get<VehicleResponse>(this.vehicleUrl).pipe(
33      map(vr => vr.results),
34      delay(2000)
35    );
36  });
37  vehicles = computed(() => this.vehiclesResource.value() ?? [] as Vehicle[]);
38  isLoading = this.vehiclesResource.isLoading;
39
40  vEff = effect(() => console.log('Selection:', this.selectedVehicle()));
41  vehiclesEff = effect(() => console.log('Vehicle data:', this.vehicles()));
42 }
43
44 export interface VehicleResponse {
45   count: number;
46   next: string;
47   previous: string;
48   results: Vehicle[];

```

Star Wars Vehicle Sales

Welcome to Star Wars Vehicle Sales



https://saprovidingservicesdeborahk-bmrk-4200-1b4252d6.local-credentialless.webcontainer.io/home

Ln 5, Col 1 Spaces: 3 UTF-8 LF typeScript Layout: US

Elements Console Performance insights Recorder Sources Network Performance Memory Application Security Lighthouse Default levels 52 Issues: 2 4 46 2 hidden

vehicle.service.ts:40 vehicle.service.ts:41 vehicle.service.ts:41 vehicle.service.ts:40 vehicle.service.ts:40

Selection: undefined

Vehicle data: ▶ []

Vehicle data: ▶ (10) [{...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}]

selection: ▶ {name: 'X-34 Landspeeder', model: 'X-34 Landspeeder', manufacturer: 'SoroSuub Corporation', cost\_in\_credits: '10550', length: '3.4', ...}

The screenshot shows a browser window with the URL [stackblitz.com/edit/providing-services-route-deborahk-wip?file=src/app/vehicles/vehicle.service.ts](https://stackblitz.com/edit/providing-services-route-deborahk-wip?file=src/app/vehicles/vehicle.service.ts). The page displays a component for selecting a vehicle. On the right, there's a modal titled "Select a Vehicle" showing an X-34 landspeeder selected. The quantity is set to 1. The vehicle details are: Name: X-34 landspeeder, Model: X-34 Landspeeder, Manufacturer: SoroSuub Corporation, Cost in Credits: 10550, and Length: 3.4 meters. Below the modal, the browser's developer tools console shows the vehicle object being logged.

```

30
31     private vehiclesResource = rxResource({
32       loader: () => this.http.get<VehicleResponse>(this.vehicleUrl).pipe(
33         map(vr => vr.results),
34         delay(2000)
35       );
36     );
37     vehicles = computed(() => this.vehiclesResource.value() ?? [] as Vehicle[]);
38     isLoading = this.vehiclesResource.isLoading;
39
40     vEff = effect(() => console.log('Selection:', this.selectedVehicle()));
41     vehiclesEff = effect(() => console.log('Vehicle data:', this.vehicles()));
42   }
43
44   export interface VehicleResponse {
45     count: number;
46     next: string;
47     previous: string;
48     results: Vehicle[];
49   }
50
51   export interface Vehicle {
52     name: string;
53     cost_in_credits: number;
54   }

```

The single instance is being reused anytime the user accesses the route in future

This screenshot is similar to the one above, but the vehicle selection dropdown in the UI is now empty. The rest of the application and the developer tools console output remain the same.

```

30
31     private vehiclesResource = rxResource({
32       loader: () => this.http.get<VehicleResponse>(this.vehicleUrl).pipe(
33         map(vr => vr.results),
34         delay(2000)
35       );
36     );
37     vehicles = computed(() => this.vehiclesResource.value() ?? [] as Vehicle[]);
38     isLoading = this.vehiclesResource.isLoading;
39
40     vEff = effect(() => console.log('Selection:', this.selectedVehicle()));
41     vehiclesEff = effect(() => console.log('Vehicle data:', this.vehicles()));
42   }
43
44   export interface VehicleResponse {
45     count: number;
46     next: string;
47     previous: string;
48     results: Vehicle[];
49   }
50
51   export interface Vehicle {
52     name: string;
53     cost_in_credits: number;
54   }

```

Next, let us now see how child routes work with our service

The screenshot shows a terminal window with the following command history:

```

~/projects/sia-providing-services-deborahk
> ng g c vehicles/vehicle-detail --inline-template --inline-style --skip-tests
CREATE src/app/vehicles/vehicle-detail/vehicle-detail.component.ts (227 bytes)

~/projects/sia-providing-services-deborahk
> ng g c vehicles/vehicle-films --inline-template --inline-style --skip-tests
CREATE src/app/vehicles/vehicle-films/vehicle-films.component.ts (224 bytes)

~/projects/sia-providing-services-deborahk
> ls

```

The screenshot shows the StackBlitz editor with the file `src/app/vehicles/vehicle-films/vehicle-films.component.ts` selected in the Explorer. The code defines a component named `VehicleFilmsComponent` with a template containing the message "vehicle-films works!". To the right, the application interface displays a "Select a Vehicle" form where "X-34 landspeeder" is selected, quantity is 1, and the total price is 10550.

```

1 import { Component } from '@angular/core';
2
3 @Component({
4   selector: 'app-vehicle-films',
5   imports: [],
6   template: `
7     <p>
8       vehicle-films works!
9     </p>
10    `,
11   styles: []
12 })
13 export class VehicleFilmsComponent {
14 }
15
16

```

This screenshot is identical to the one above, showing the `vehicle-films` component code and the application interface displaying the vehicle selection form.

```

1 import { Component } from '@angular/core';
2
3 @Component({
4   selector: 'app-vehicle-films',
5   imports: [],
6   template: `
7     <p>
8       vehicle-films works!
9     </p>
10    `,
11   styles: []
12 })
13 export class VehicleFilmsComponent {
14 }
15
16

```

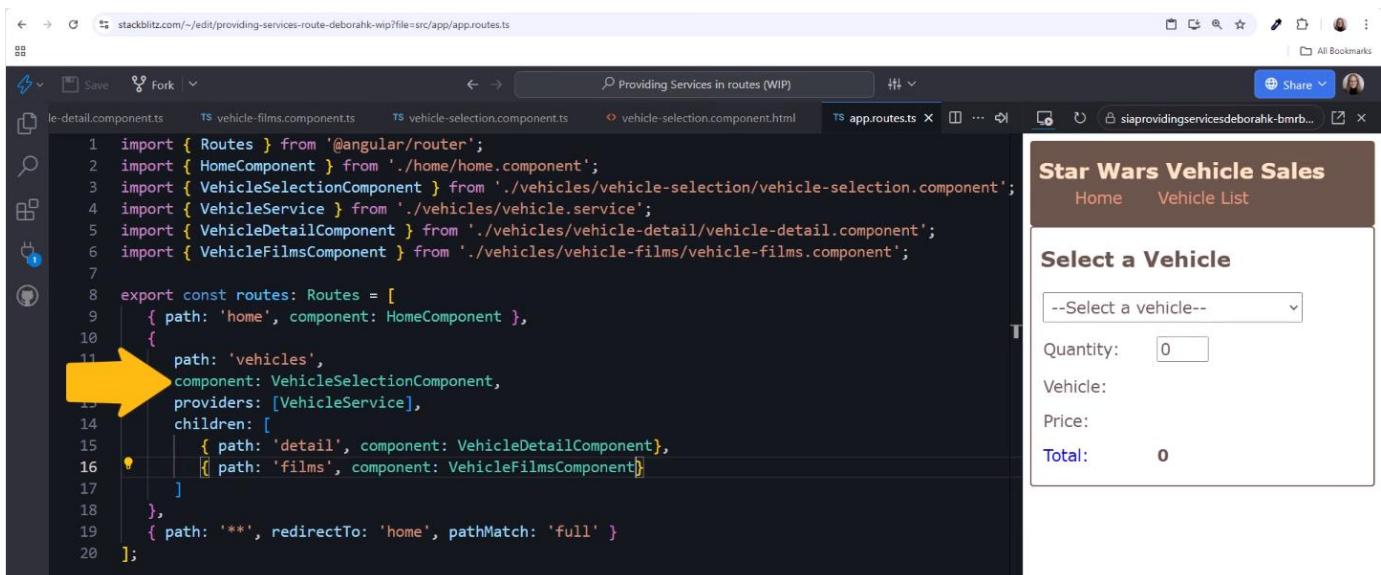
The screenshot shows the `app.routes.ts` file in the StackBlitz editor. It defines a routes array with two entries: a home route and a vehicles route. The vehicles route has a provider named `VehicleService`. To the right, the application interface shows the vehicle selection form with "X-34 landspeeder" selected, quantity 1, and total 10550.

```

1 import { Routes } from '@angular/router';
2 import { HomeComponent } from './home/home.component';
3 import { VehicleSelectionComponent } from './vehicles/vehicle-selection/vehicle-selection.component';
4 import { VehicleService } from './vehicles/vehicle.service';
5
6 export const routes: Routes = [
7   { path: 'home', component: HomeComponent },
8   {
9     path: 'vehicles',
10    component: VehicleSelectionComponent,
11    providers: [VehicleService]
12   },
13   { path: '**', redirectTo: 'home', pathMatch: 'full' }
14 ];

```

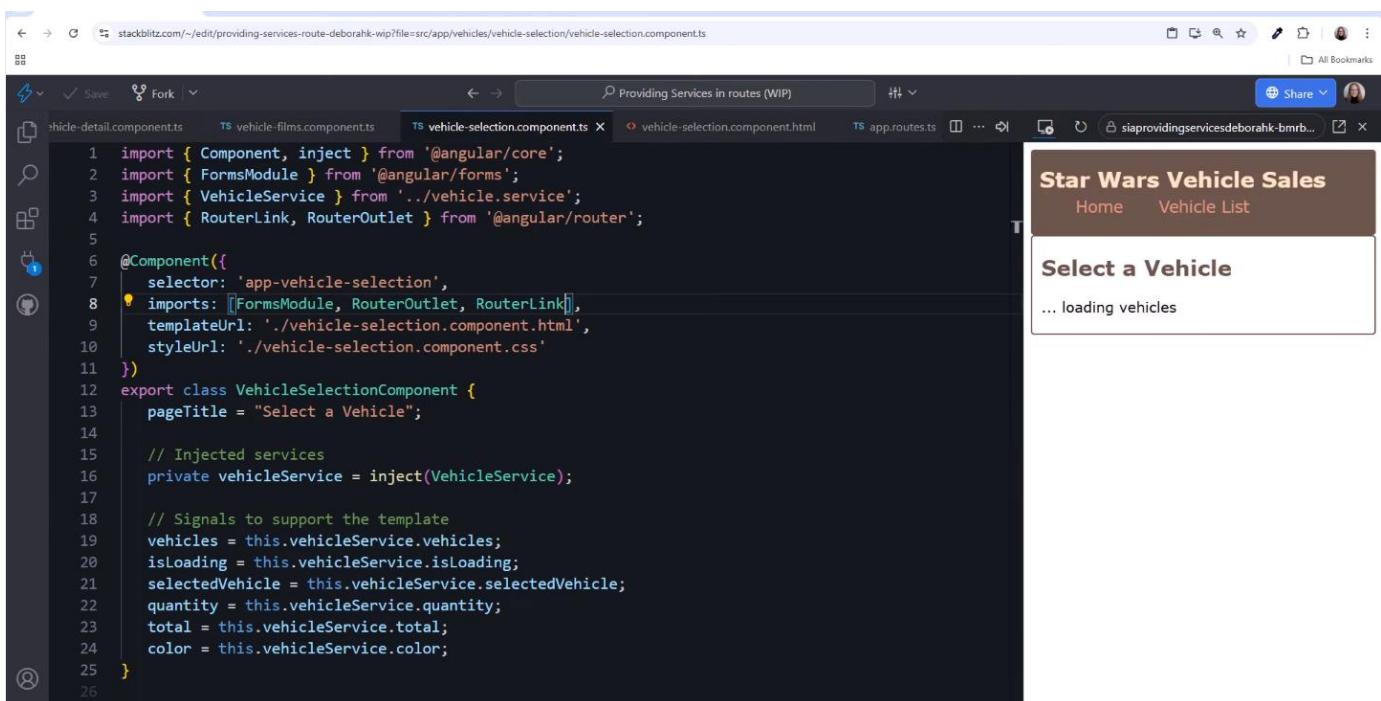
Next, we need to add child routes for the 2 new child components as below



```

1 import { Routes } from '@angular/router';
2 import { HomeComponent } from './home/home.component';
3 import { VehicleSelectionComponent } from './vehicles/vehicle-selection/vehicle-selection.component';
4 import { VehicleService } from './vehicles/vehicle.service';
5 import { VehicleDetailComponent } from './vehicles/vehicle-detail/vehicle-detail.component';
6 import { VehicleFilmsComponent } from './vehicles/vehicle-films/vehicle-films.component';
7
8 export const routes: Routes = [
9   { path: 'home', component: HomeComponent },
10  {
11    path: 'vehicles',
12    component: VehicleSelectionComponent,
13    providers: [VehicleService],
14    children: [
15      { path: 'detail', component: VehicleDetailComponent},
16      { path: 'films', component: VehicleFilmsComponent}
17    ],
18  },
19  { path: '**', redirectTo: 'home', pathMatch: 'full' }
20 ];

```



```

1 import { Component, inject } from '@angular/core';
2 import { FormsModule } from '@angular/forms';
3 import { VehicleService } from '../vehicle.service';
4 import { RouterLink, RouterOutlet } from '@angular/router';
5
6 @Component({
7   selector: 'app-vehicle-selection',
8   imports: [FormsModule, RouterOutlet, RouterLink],
9   templateUrl: './vehicle-selection.component.html',
10  styleUrls: ['./vehicle-selection.component.css'
11 })
12 export class VehicleSelectionComponent {
13   pageTitle = "Select a Vehicle";
14
15   // Injected services
16   private vehicleService = inject(VehicleService);
17
18   // Signals to support the template
19   vehicles = this.vehicleService.vehicles;
20   isLoading = this.vehicleService.isLoading;
21   selectedVehicle = this.vehicleService.selectedVehicle;
22   quantity = this.vehicleService.quantity;
23   total = this.vehicleService.total;
24   color = this.vehicleService.color;
25 }

```

We then add the **RouterOutlet** and the **Routerlink** as above

stackblitz.com/~edit/providing-services-route-deborahk-wip?file=src/app/vehicles/vehicle-selection/vehicle-selection.component.html

```
1 <div class="content">
2   <div class="title">
3     {{pageTitle}}
4   </div>
5
6   @if (isLoading()) {
7     <div>... loading vehicles</div>
8   } @else {
9     <div class="grid">
10       <div class="row">
11         <div class="cellFull">
12           <select [(ngModel)]="selectedVehicle">
13             <option value="undefined" disabled selected>--Select a vehicle--</option>
14             @for(vehicle of vehicles()); track vehicle) {
15               <option [ngValue]='vehicle'>{{ vehicle.name }}</option>
16             }
17           </select></div>
18         </div>
19         <div class="row">
20           <div class="cellLeft"><label for='quantity'>Quantity:</label></div>
21           <div class="cellRight"><input class="quantityInput" id='quantity' type='number' [(ngModel)]="quantity" value="0" /></div>
22         </div>
23         <div class="row">
24           <div class="cellLeft">Vehicle:</div>
25           <div class="cellRight">{{ selectedVehicle()?.name }}</div>
26         </div>
27         <div class="row">
28           <div class="cellLeft">Price:</div>
29           <div class="cellRight">{{ selectedVehicle()?.cost_in_credits }}</div>
30         </div>
31         <div class="row">
32           <div class="cellLeft" [style.color]:'color()'>Total:</div>
33           <div class="cellRight">{{ total() }}</div>
34         </div>
35       </div>
36     </div>
37   @if (selectedVehicle()) {
38     <div>
39       <a [routerLink]=['detail']>More Detail</a>
40       <a [routerLink]=['films']>Seen in these films</a>
41     </div>
42   </div>
43 }
44 }
```

stackblitz.com/~edit/providing-services-route-deborahk-wip?file=src/app/vehicles/vehicle-selection/vehicle-selection.component.html

```
21   <div class="cellRight"><input class="quantityInput" id='quantity' type='number' [(ngModel)]="quantity" value="0" /></div>
22 </div>
23 <div class="row">
24   <div class="cellLeft">Vehicle:</div>
25   <div class="cellRight">{{ selectedVehicle()?.name }}</div>
26 </div>
27 <div class="row">
28   <div class="cellLeft">Price:</div>
29   <div class="cellRight">{{ selectedVehicle()?.cost_in_credits }}</div>
30 </div>
31 <div class="row">
32   <div class="cellLeft" [style.color]:'color()'>Total:</div>
33   <div class="cellRight">{{ total() }}</div>
34 </div>
35 </div>
36 </div>
37 @if (selectedVehicle()) {
38   <div>
39     <a [routerLink]=['detail']>More Detail</a>
40     <a [routerLink]=['films']>Seen in these films</a>
41   </div>
42 </div>
43 }
44 }
```

Providing Services in routes (WIP)

```

21   <div class="cellRight"><input class="quantityInput" id='quantity' type='number' [(ngModel)]=>
22   </div>
23   <div class="row">
24     <div class="cellLeft">Vehicle:</div>
25     <div class="cellRight">{{ selectedVehicle()?.name }}</div>
26   </div>
27   <div class="row">
28     <div class="cellLeft">Price:</div>
29     <div class="cellRight">{{ selectedVehicle()?.cost_in_credits }}</div>
30   </div>
31   <div class="row">
32     <div class="cellLeft" [style.color]="'color()">'Total: </div>
33     <div class="cellRight">{{ total() }}</div>
34   </div>
35 </div>
36 }
37 </div>
38 @if (selectedVehicle()) {
39   <div>
40     <div><a [routerLink]=["['detail']]">More Detail</a></div>
41     <div><a [routerLink]=["['films']]">Seen in these films</a></div>
42     <div><router-outlet/></div>
43   </div>
44 }
45

```

**Star Wars Vehicle Sales**

Select a Vehicle

- Select a vehicle--
- Select a vehicle--
- Sand Crawler
- T-16 skyhopper**
- X-34 landspeeder
- TIE/LN starfighter
- Snowspeeder
- TIE bomber
- AT-AT
- AT-ST
- Storm IV Twin-Pod cloud car
- Sail barge

Providing Services in routes (WIP)

```

21   <div class="cellRight"><input class="quantityInput" id='quantity' type='number' [(ngModel)]=>
22   </div>
23   <div class="row">
24     <div class="cellLeft">Vehicle:</div>
25     <div class="cellRight">{{ selectedVehicle()?.name }}</div>
26   </div>
27   <div class="row">
28     <div class="cellLeft">Price:</div>
29     <div class="cellRight">{{ selectedVehicle()?.cost_in_credits }}</div>
30   </div>
31   <div class="row">
32     <div class="cellLeft" [style.color]="'color()">'Total: </div>
33     <div class="cellRight">{{ total() }}</div>
34   </div>
35 </div>
36 }
37 </div>
38 @if (selectedVehicle()) {
39   <div>
40     <div><a [routerLink]=["['detail']]">More Detail</a></div>
41     <div><a [routerLink]=["['films']]">Seen in these films</a></div>
42     <div><router-outlet/></div>
43   </div>
44 }
45

```

**Star Wars Vehicle Sales**

Select a Vehicle

X-34 landspeeder

Quantity:

Vehicle: **X-34 landspeeder**

Price: **10550**

Total: **10550**

More Detail  
Seen in these films



stackblitz.com/~edit/providing-services-route-deborahk-wip?file=src/app/vehicles/vehicle-selection/vehicle-selection.component.html

Providing Services in routes (WIP)

vehicle-detail.component.ts    vehicle-films.component.ts    vehicle-selection.component.ts    vehicle-selection.component.html    app.routes.ts

```

21   |     <div class="cellRight"><input class="quantityInput" id='quantity' type='number' [(ngModel)]="selectedVehicle">
22   |   </div>
23   |   <div class="row">
24   |     <div class="cellLeft">Vehicle:</div>
25   |     <div class="cellRight">{{ selectedVehicle()?.name }}</div>
26   |   </div>
27   |   <div class="row">
28   |     <div class="cellLeft">Price:</div>
29   |     <div class="cellRight">{{ selectedVehicle()?.cost_in_credits }}</div>
30   |   </div>
31   |   <div class="row">
32   |     <div class="cellLeft" [style.color]="'color()">'Total: </div>
33   |     <div class="cellRight">{{ total() }}</div>
34   |   </div>
35   | </div>
36   }
37 </div>
38 @if (selectedVehicle()) {
39   <div>
40     <a [routerLink]=["['detail']]">More Detail</a></div>
41     <div><a [routerLink]=["['films']]">Seen in these films</a></div>
42     <div><router-outlet></div>
43   </div>
44 }

```

**Star Wars Vehicle Sales**

Home    Vehicle List

Select a Vehicle

X-34 landspeeder

Quantity: 1

Vehicle: X-34 landspeeder

Price: 10550

Total: 10550

More Detail  
Seen in these films

vehicle-detail works!

stackblitz.com/~edit/providing-services-route-deborahk-wip?file=src/app/vehicles/vehicle-selection/vehicle-selection.component.html

Providing Services in routes (WIP)

vehicle-detail.component.ts    vehicle-films.component.ts    vehicle-selection.component.ts    vehicle-selection.component.html    app.routes.ts

```

21   |     <div class="cellRight"><input class="quantityInput" id='quantity' type='number' [(ngModel)]="selectedVehicle">
22   |   </div>
23   |   <div class="row">
24   |     <div class="cellLeft">Vehicle:</div>
25   |     <div class="cellRight">{{ selectedVehicle()?.name }}</div>
26   |   </div>
27   |   <div class="row">
28   |     <div class="cellLeft">Price:</div>
29   |     <div class="cellRight">{{ selectedVehicle()?.cost_in_credits }}</div>
30   |   </div>
31   |   <div class="row">
32   |     <div class="cellLeft" [style.color]="'color()">'Total: </div>
33   |     <div class="cellRight">{{ total() }}</div>
34   |   </div>
35   | </div>
36   }
37 </div>
38 @if (selectedVehicle()) {
39   <div>
40     <a [routerLink]=["['detail']]">More Detail</a></div>
41     <div><a [routerLink]=["['films']]">Seen in these films</a></div>
42     <div><router-outlet></div>
43   </div>
44 }

```

**Star Wars Vehicle Sales**

Home    Vehicle List

Select a Vehicle

X-34 landspeeder

Quantity: 1

Vehicle: X-34 landspeeder

Price: 10550

Total: 10550

More Detail  
Seen in these films

vehicle-films works!

Next, let us inject the service into each child component

The screenshot shows the StackBlitz editor with the file `vehicle-detail.component.ts` open. The code defines a simple component that displays a message: "vehicle-detail works!". To the right, the rendered application interface titled "Star Wars Vehicle Sales" shows a dropdown menu set to "X-34 landspeeder", a quantity of "1", and a total price of "10550". Below the dropdown, there are links for "More Detail" and "Seen in these films". A footer message "vehicle-films works!" is visible.

```
1 import { Component } from '@angular/core';
2
3 @Component({
4   selector: 'app-vehicle-detail',
5   imports: [],
6   template: `
7     <p>
8       vehicle-detail works!
9     </p>
10    `,
11   styles: []
12 })
13 export class VehicleDetailComponent {
14
15 }
```

The screenshot shows the StackBlitz editor with the file `vehicle-detail.component.ts` open. The code has been updated to inject the `VehicleService` and access the selected vehicle's name via `this.vehicleService.selectedVehicle.name`. The rendered application interface shows a loading message "... loading vehicles".

```
1 import { Component, inject } from '@angular/core';
2 import { VehicleService } from '../vehicle.service';
3
4 @Component({
5   selector: 'app-vehicle-detail',
6   imports: [],
7   template: `
8     <p>
9       Details for vehicle {{ selectedVehicle()?.name }}
10    </p>
11    `,
12   styles: []
13 })
14 export class VehicleDetailComponent {
15   private vehicleService = inject(VehicleService);
16
17   selectedVehicle = this.vehicleService.selectedVehicle;
18 }
```

The screenshot shows the StackBlitz editor with the file `vehicle-films.component.ts` open. The code defines a component that displays a message: "Films for vehicle {{ selectedVehicle()?.name }}". To the right, the rendered application interface titled "Star Wars Vehicle Sales" shows a dropdown menu labeled "--Select a vehicle--", a quantity of "0", and a total price of "0".

```
1 import { Component, inject } from '@angular/core';
2 import { VehicleService } from '../vehicle.service';
3
4 @Component({
5   selector: 'app-vehicle-films',
6   imports: [],
7   template: `
8     <p>
9       Films for vehicle {{ selectedVehicle()?.name }}
10    </p>
11    `,
12   styles: []
13 })
14 export class VehicleFilmsComponent {
15   private vehicleService = inject(VehicleService);
16
17   selectedVehicle = this.vehicleService.selectedVehicle;
18 }
```

We can now see how our injected service works below

Providing Services in routes (WIP)

```
1 import { Component, inject } from '@angular/core';
2 import { VehicleService } from '../vehicle.service';
3
4 @Component({
5   selector: 'app-vehicle-films',
6   imports: [],
7   template: `
8     <p>
9       Films for vehicle {{ selectedVehicle()?.name }}
10    </p>
11  `,
12  styles: []
13})
14 export class VehicleFilmsComponent {
15   private vehicleService = inject(VehicleService);
16
17   selectedVehicle = this.vehicleService.selectedVehicle;
18
19}
```

Star Wars Vehicle Sales

Welcome to Star Wars Vehicle Sales



YouTube Channel

Providing Services in routes (WIP)

```
1 import { Component, inject } from '@angular/core';
2 import { VehicleService } from '../vehicle.service';
3
4 @Component({
5   selector: 'app-vehicle-films',
6   imports: [],
7   template: `
8     <p>
9       Films for vehicle {{ selectedVehicle()?.name }}
10    </p>
11  `,
12  styles: []
13})
14 export class VehicleFilmsComponent {
15   private vehicleService = inject(VehicleService);
16
17   selectedVehicle = this.vehicleService.selectedVehicle;
18
19}
```

Star Wars Vehicle Sales

Home **Vehicle List**

Select a Vehicle

--Select a vehicle--

Quantity: 0

Vehicle:

Price:

Total: 0

Providing Services in routes (WIP)

```
1 import { Component, inject } from '@angular/core';
2 import { VehicleService } from '../vehicle.service';
3
4 @Component({
5   selector: 'app-vehicle-films',
6   imports: [],
7   template: `
8     <p>
9       Films for vehicle {{ selectedVehicle()?.name }}
10    </p>
11  `,
12  styles: []
13})
14 export class VehicleFilmsComponent {
15   private vehicleService = inject(VehicleService);
16
17   selectedVehicle = this.vehicleService.selectedVehicle;
18
19}
```

Star Wars Vehicle Sales

Home **Vehicle List**

Select a Vehicle

--Select a vehicle--

--Select a vehicle--

- Sand Crawler
- T-16 skyhopper
- X-34 landspeeder**
- TIE/LN starfighter
- Snowspeeder
- TIE bomber
- AT-AT
- AT-ST
- Storm IV Twin-Pod cloud car
- Sail barge

The screenshot shows a StackBlitz editor with three tabs: `vehicle-detail.component.ts`, `vehicle-films.component.ts`, and `vehicle-selection.component.ts`. The `vehicle-films.component.ts` tab is active, displaying the following code:

```
1 import { Component, inject } from '@angular/core';
2 import { VehicleService } from '../vehicle.service';
3
4 @Component({
5   selector: 'app-vehicle-films',
6   imports: [],
7   template: `
8     <p>
9       Films for vehicle {{ selectedVehicle()?.name }}
10    </p>
11  `,
12   styles: []
13 })
14 export class VehicleFilmsComponent {
15   private vehicleService = inject(VehicleService);
16
17   selectedVehicle = this.vehicleService.selectedVehicle;
18
19 }
```

To the right, a browser window displays the application's interface. It has a header "Star Wars Vehicle Sales" with "Home" and "Vehicle List" links. Below is a section titled "Select a Vehicle" with a dropdown set to "X-34 landspeeder", a quantity input (1), and vehicle details: Price: 10550, Total: 10550. A link "Seen in these films" is visible.

The screenshot shows the same StackBlitz editor and browser setup. The `vehicle-films.component.ts` code remains the same as above. In the browser, the "Select a Vehicle" section now includes a "Details for vehicle X-34 landspeeder" message, indicating that the vehicle selection logic has been moved to the parent component.

Our parent and child components are now sharing the same `VehicleService` instance and thus the same state

The screenshot shows the final state of the application. The `vehicle-films.component.ts` code is identical to the previous screenshots. In the browser, the "Select a Vehicle" section now includes a "Films for vehicle X-34 landspeeder" message, indicating that the child component no longer handles vehicle selection.

Provide in	Initialization	Scope	Lifetime
"root"	One instance (singleton)	State shared with all components and services in the application	Life of the application
route	One instance	State shared with components on the route or child routes	Life of the application
component	One instance for each component	State shared with child components	Life of the component that injected it