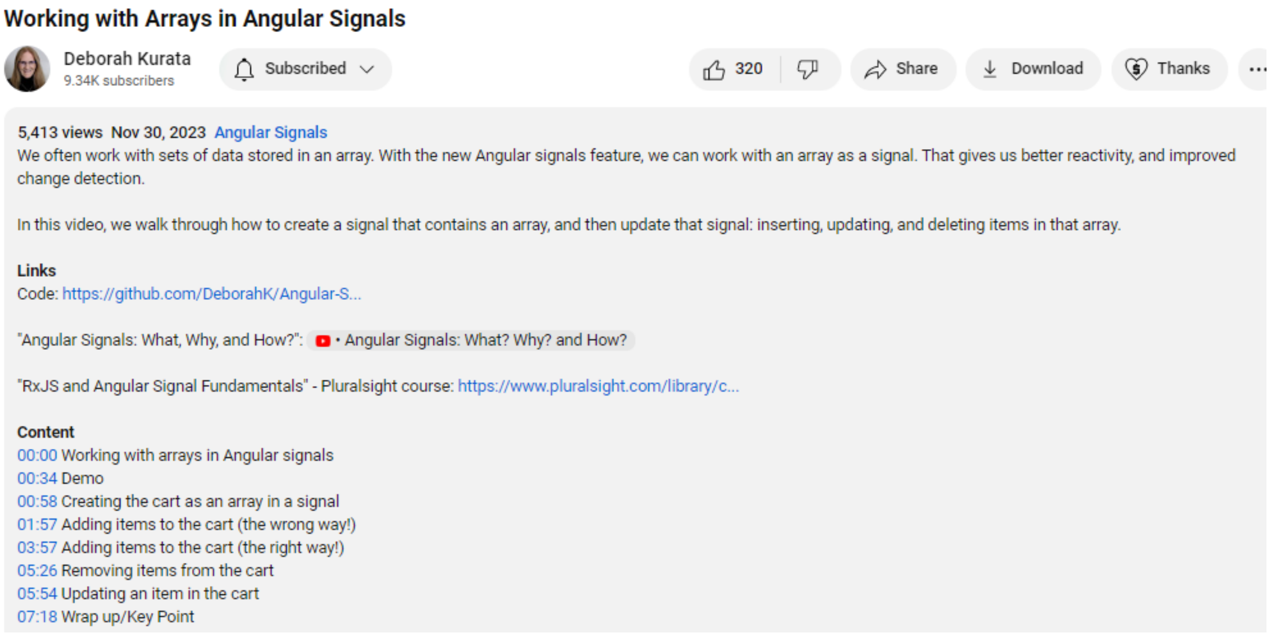


Working with Arrays in Angular Signals

Deborah Kurata
9.34K subscribers

5,413 views Nov 30, 2023 Angular Signals

We often work with sets of data stored in an array. With the new Angular signals feature, we can work with an array as a signal. That gives us better reactivity, and improved change detection.

In this video, we walk through how to create a signal that contains an array, and then update that signal: inserting, updating, and deleting items in that array.

Links

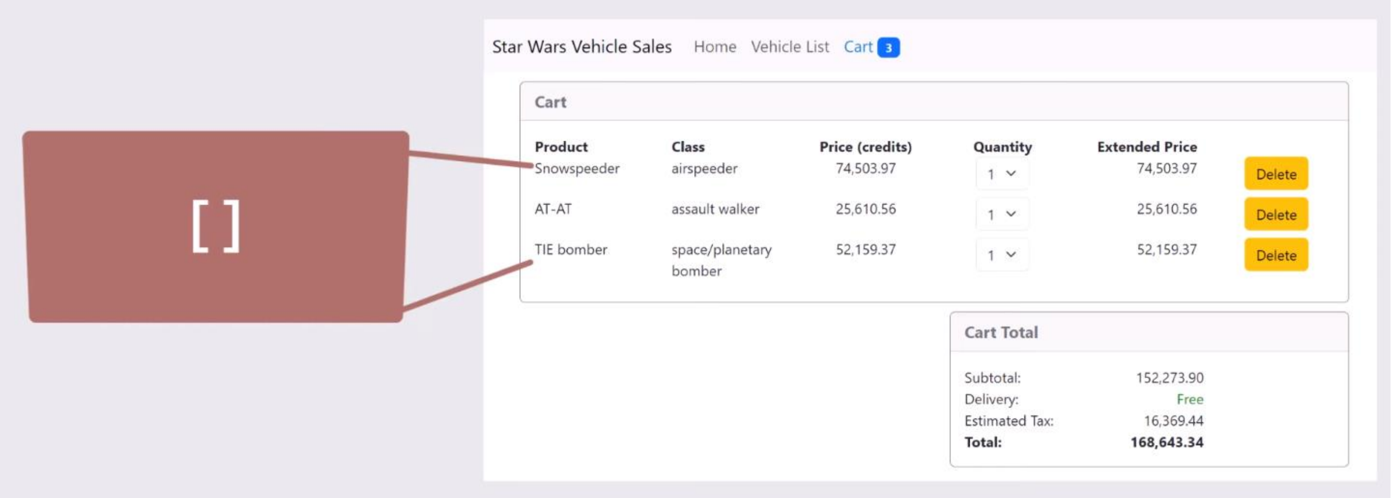
Code: https://github.com/DeborahK/Angular-S...

"Angular Signals: What, Why, and How?": • Angular Signals: What? Why? and How?

"RxJS and Angular Signal Fundamentals" - Pluralsight course: https://www.pluralsight.com/library/c...

Content

00:00 Working with arrays in Angular signals
00:34 Demo
00:58 Creating the cart as an array in a signal
01:57 Adding items to the cart (the wrong way!)
03:57 Adding items to the cart (the right way!)
05:26 Removing items from the cart
05:54 Updating an item in the cart
07:18 Wrap up/Key Point



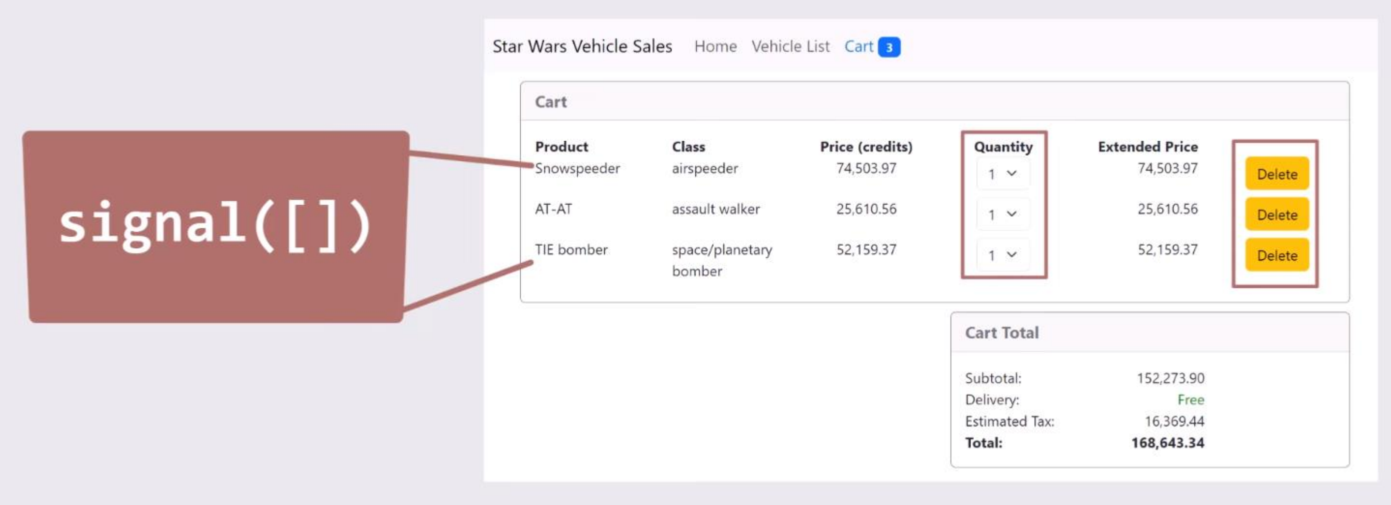
[]

Cart

Product	Class	Price (credits)	Quantity	Extended Price	
Snowspeeder	airspeeder	74,503.97	1	74,503.97	Delete
AT-AT	assault walker	25,610.56	1	25,610.56	Delete
TIE bomber	space/planetary bomber	52,159.37	1	52,159.37	Delete

Cart Total

Subtotal:	152,273.90
Delivery:	Free
Estimated Tax:	16,369.44
Total:	**168,643.34**



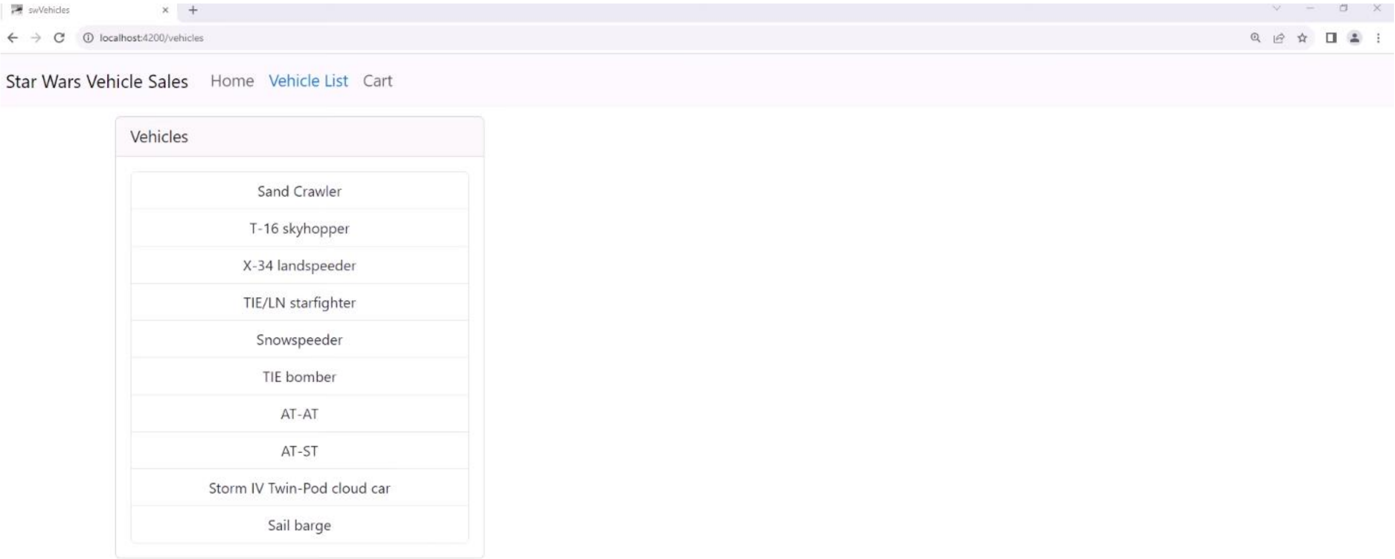
signal([])

Cart

Product	Class	Price (credits)	Quantity	Extended Price	
Snowspeeder	airspeeder	74,503.97	1	74,503.97	Delete
AT-AT	assault walker	25,610.56	1	25,610.56	Delete
TIE bomber	space/planetary bomber	52,159.37	1	52,159.37	Delete

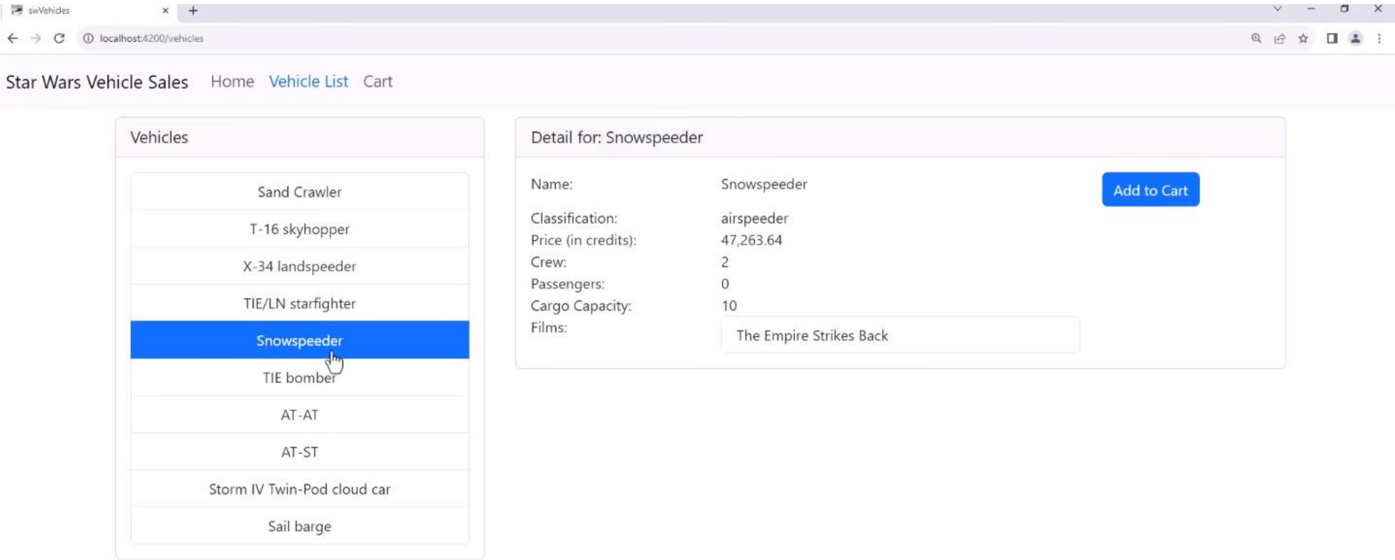
Cart Total

Subtotal:	152,273.90
Delivery:	Free
Estimated Tax:	16,369.44
Total:	**168,643.34**



Vehicles

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

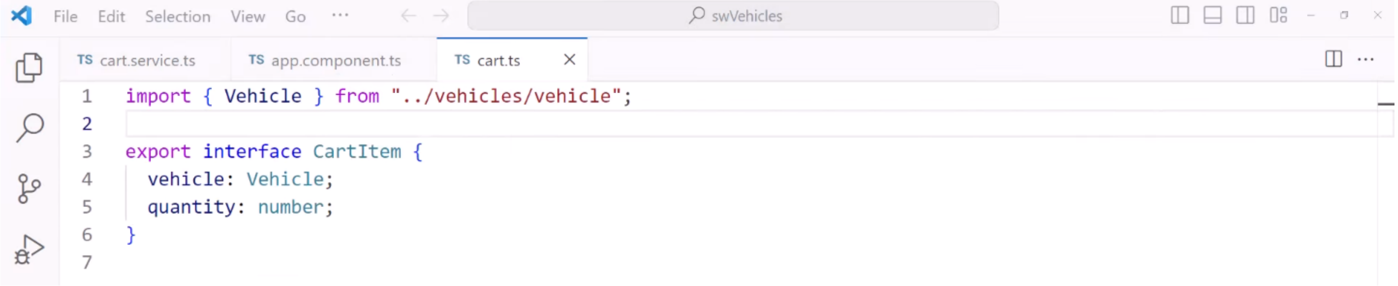


Vehicles

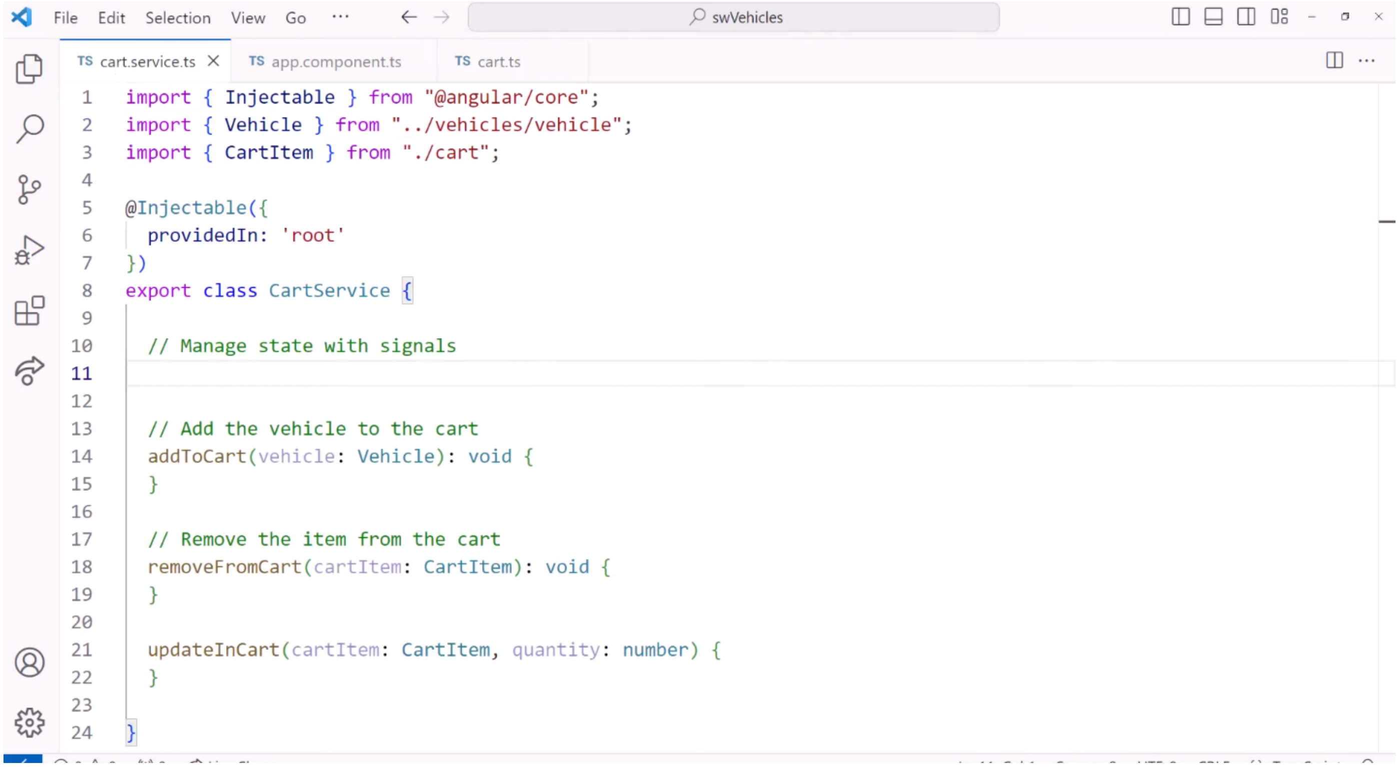
Sand Crawler
T-16 skyhopper
X-34 landspeeder
TIE/LN starfighter
Snowspeeder
TIE bomber
AT-AT
AT-ST
Storm IV Twin-Pod cloud car
Sail barge

Detail for: Snowspeeder

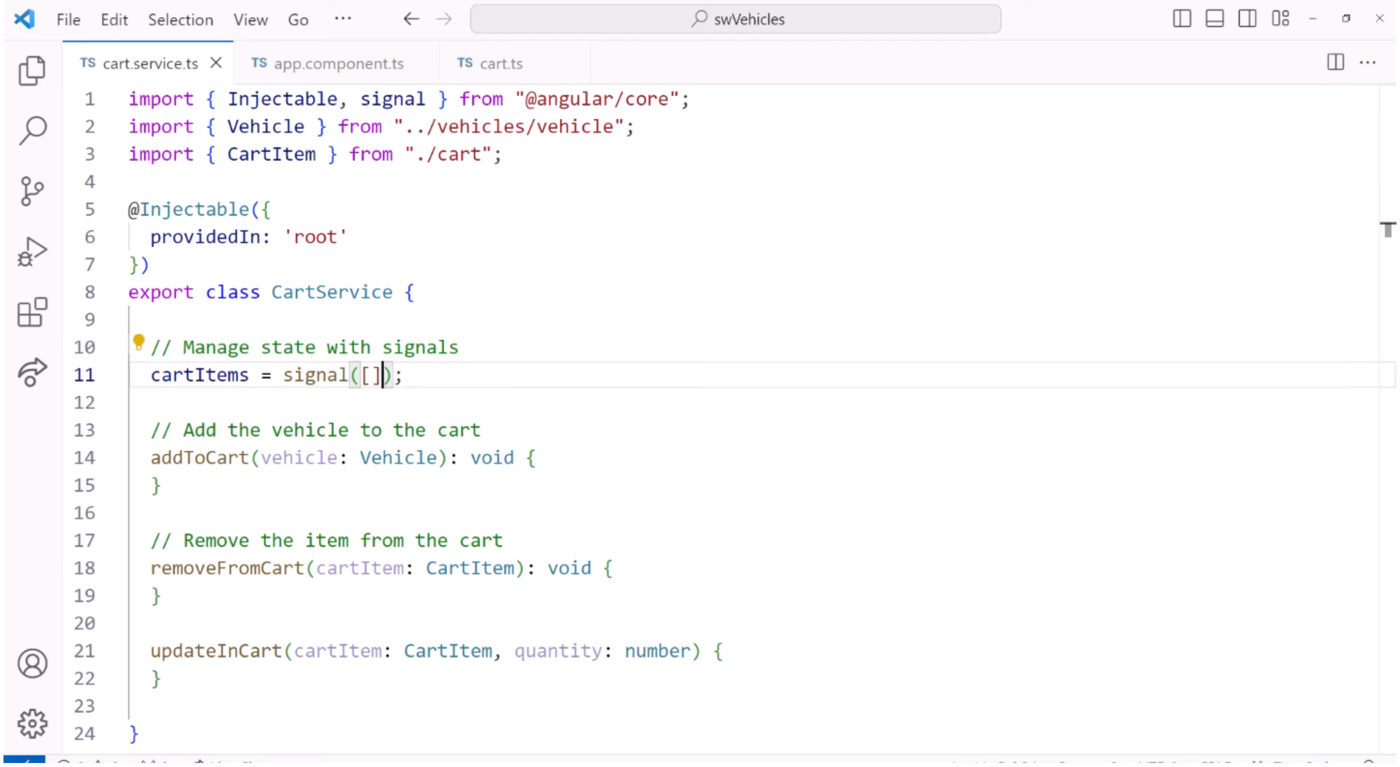
Name: Snowspeeder
Classification: airspeeder
Price (in credits): 47,263.64
Crew: 2
Passengers: 0
Cargo Capacity: 10
Films: The Empire Strikes Back



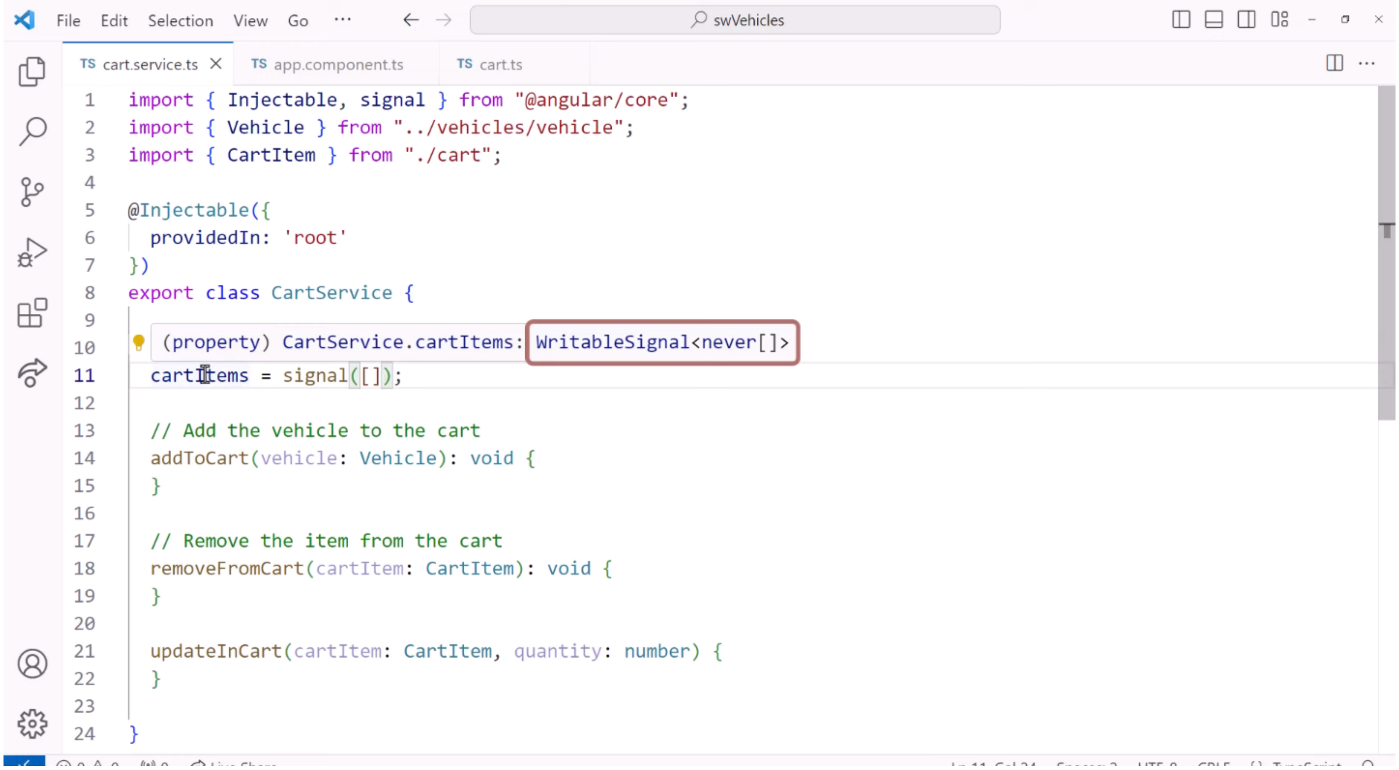
```
import { Vehicle } from "../vehicles/vehicle";

export interface CartItem {
  vehicle: Vehicle;
  quantity: number;
}

```



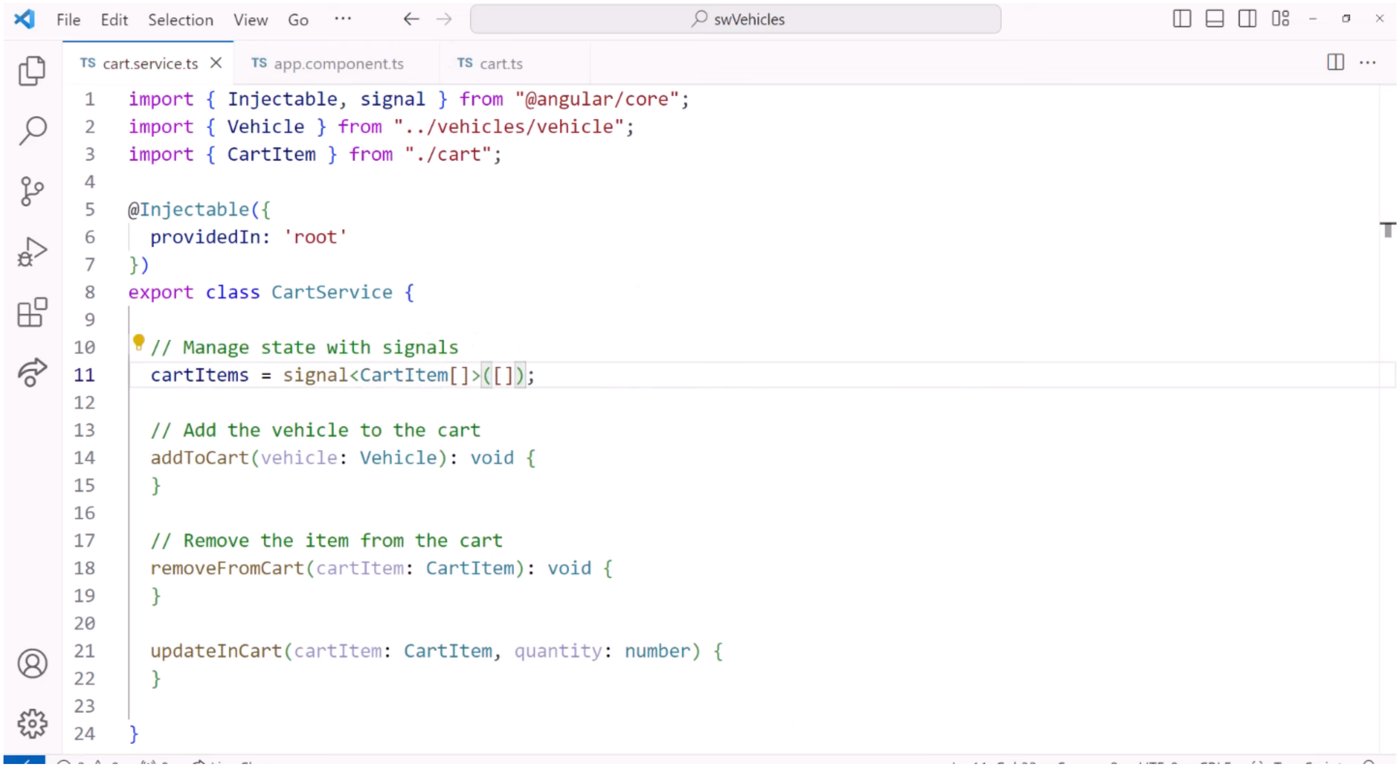
```typescript
import { Injectable } from "@angular/core";
import { Vehicle } from "../vehicles/vehicle";
import { CartItem } from "./cart";

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

  // Manage state with signals


  // Add the vehicle to the cart
  addToCart(vehicle: Vehicle): void {
  }

  // Remove the item from the cart
  removeFromCart(cartItem: CartItem): void {
  }

  updateInCart(cartItem: CartItem, quantity: number) {
  }

}
```



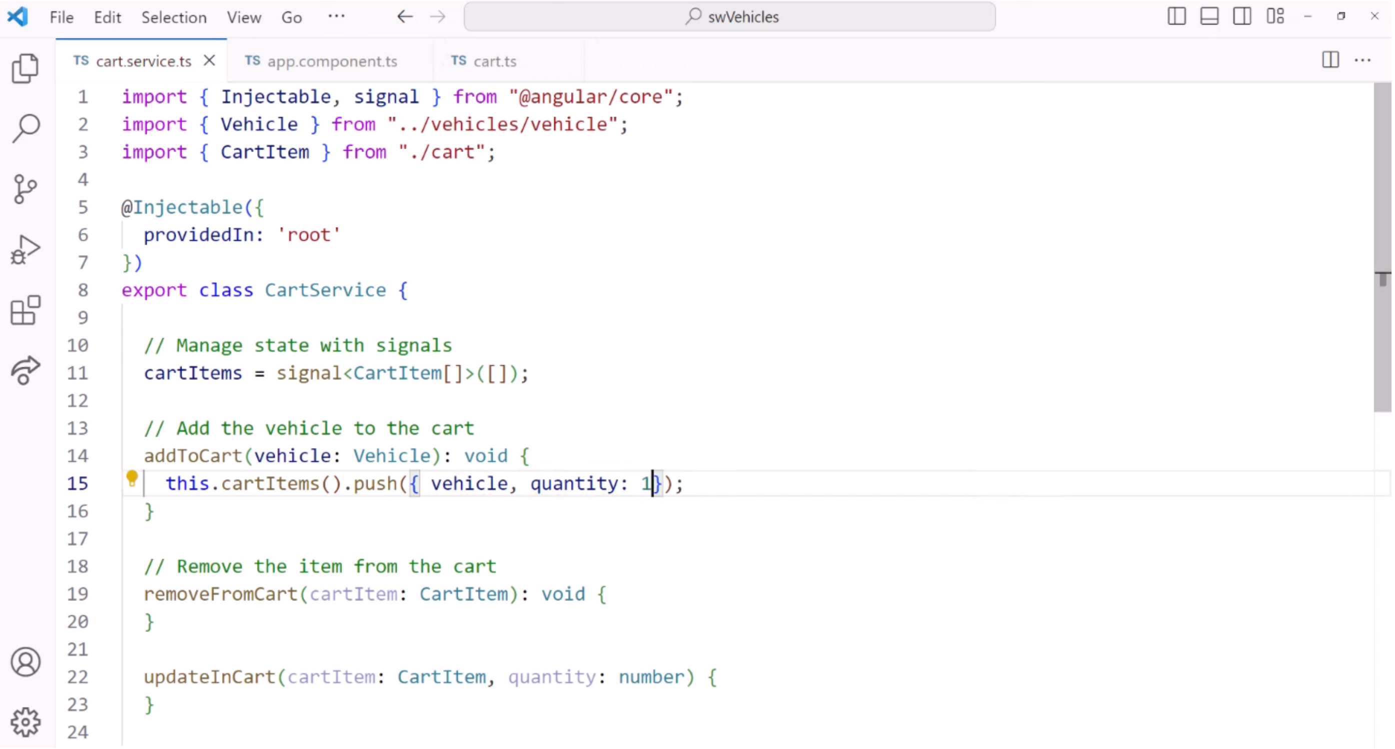
```typescript
import { Injectable, signal } from "@angular/core";
import { Vehicle } from "../vehicles/vehicle";
import { CartItem } from "./cart";

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

  // Manage state with signals
  cartItems = signal([]);

  // Add the vehicle to the cart
  addToCart(vehicle: Vehicle): void {
  }

  // Remove the item from the cart
  removeFromCart(cartItem: CartItem): void {
  }

  updateInCart(cartItem: CartItem, quantity: number) {
  }

}
```



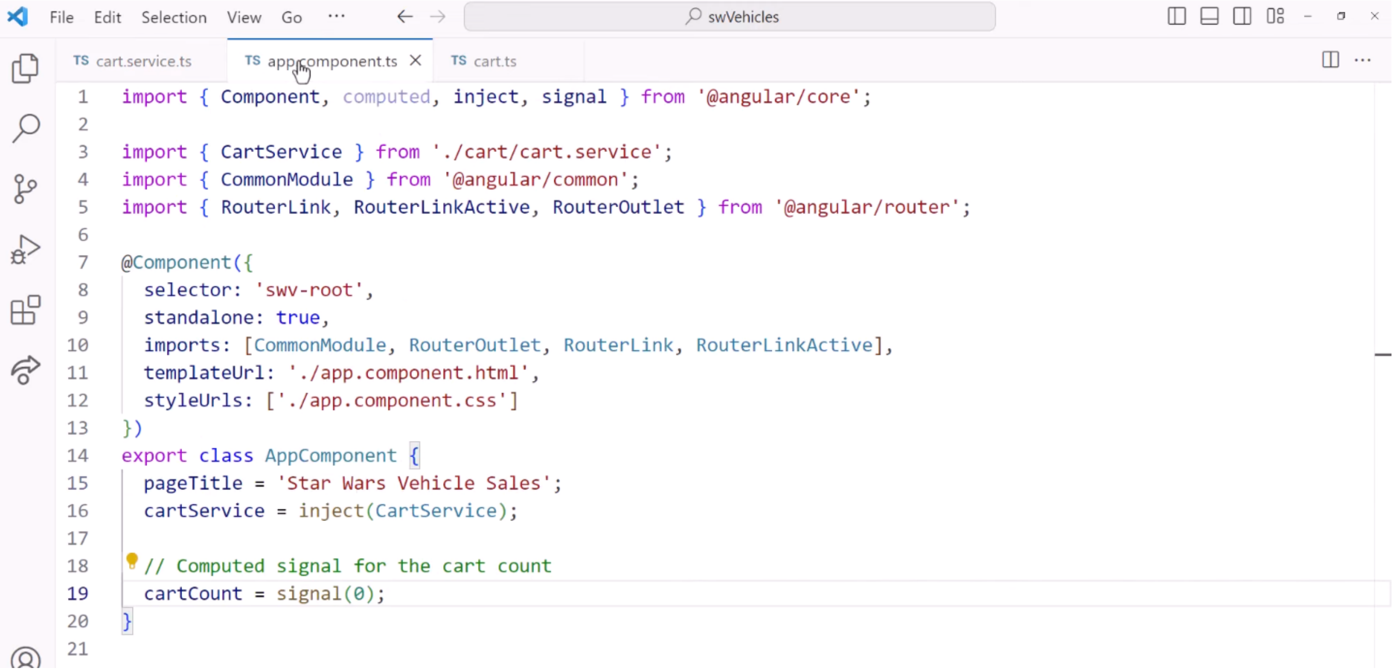
```
import { Injectable, signal } from "@angular/core";
import { Vehicle } from "../vehicles/vehicle";
import { CartItem } from "./cart";

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

  (property) CartService.cartItems: WritableSignal<never[]>
  cartItems = signal([]);

  // Add the vehicle to the cart
  addToCart(vehicle: Vehicle): void {
  }

  // Remove the item from the cart
  removeFromCart(cartItem: CartItem): void {
  }

  updateInCart(cartItem: CartItem, quantity: number) {
  }

}
```



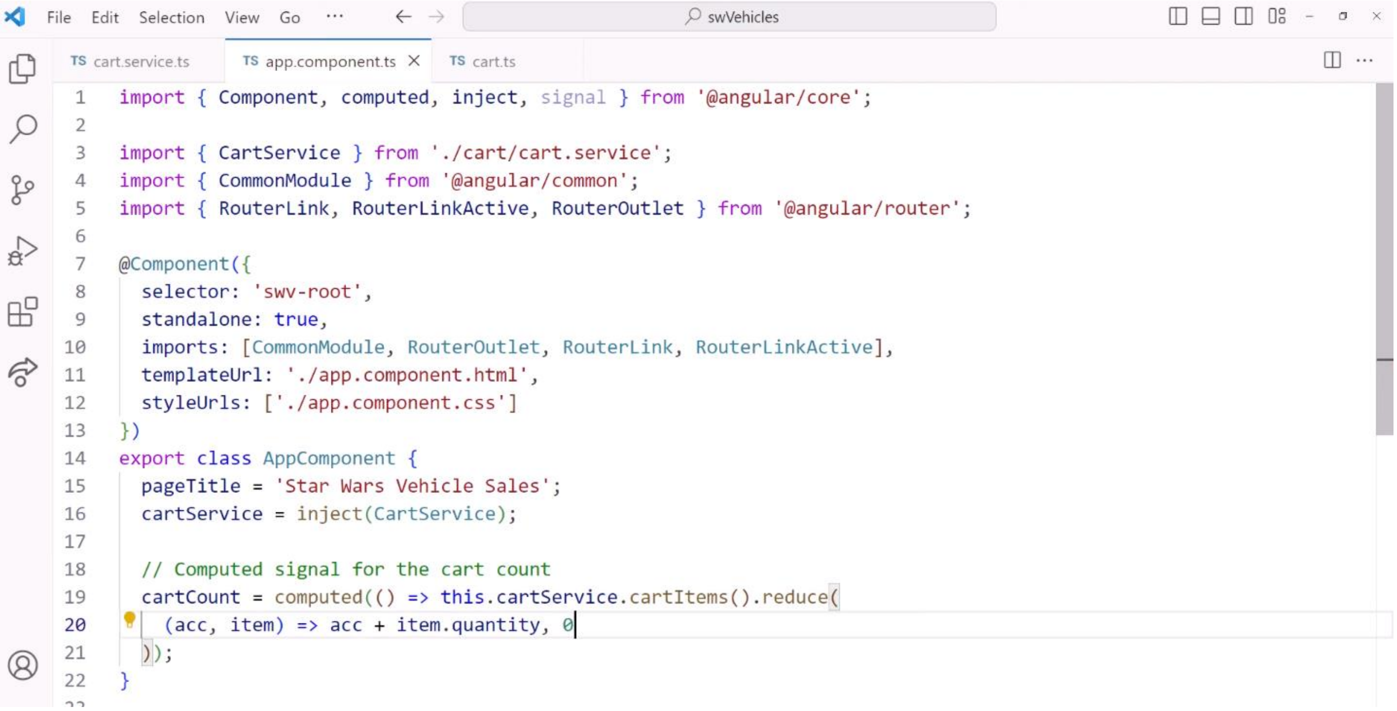
```
import { Injectable, signal } from "@angular/core";
import { Vehicle } from "../vehicles/vehicle";
import { CartItem } from "./cart";

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

  // Manage state with signals
  cartItems = signal<CartItem[]>([]);

  // Add the vehicle to the cart
  addToCart(vehicle: Vehicle): void {
  }

  // Remove the item from the cart
  removeFromCart(cartItem: CartItem): void {
  }

  updateInCart(cartItem: CartItem, quantity: number) {
  }

}
```



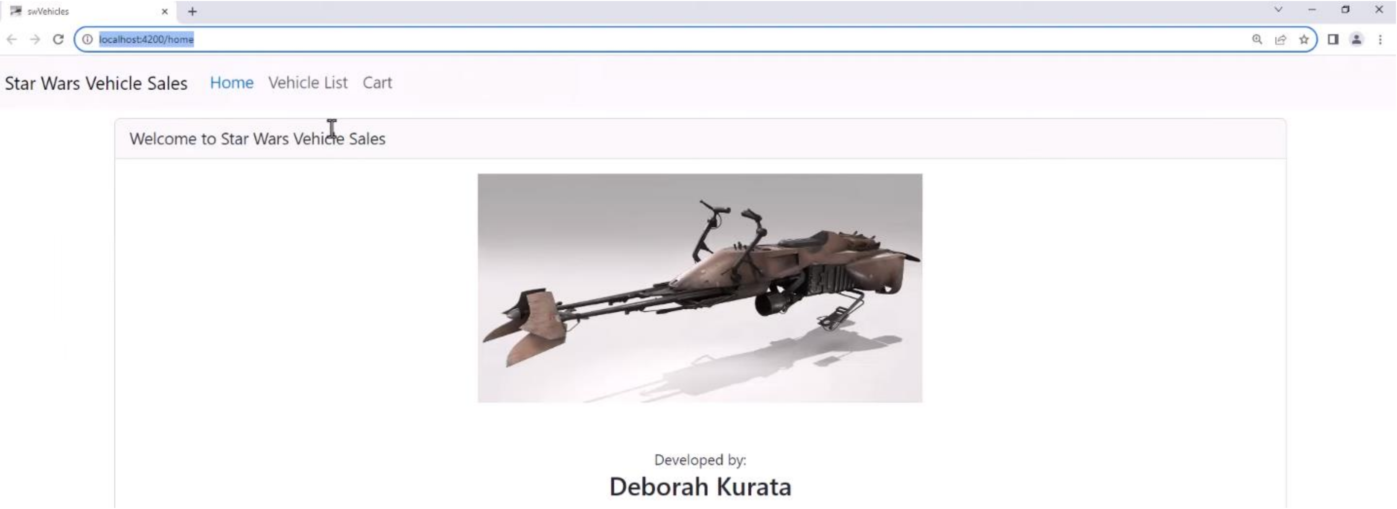
```typescript
import { Injectable, signal } from "@angular/core";
import { Vehicle } from "../vehicles/vehicle";
import { CartItem } from "./cart";

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

  // Manage state with signals
  cartItems = signal<CartItem[]>([]);

  // Add the vehicle to the cart
  addToCart(vehicle: Vehicle): void {
    this.cartItems().push({ vehicle, quantity: 1});
  }

  // Remove the item from the cart
  removeFromCart(cartItem: CartItem): void {
  }

  updateInCart(cartItem: CartItem, quantity: number) {
  }

```



```typescript
import { Component, computed, inject, signal } from '@angular/core';

import { CartService } from './cart/cart.service';
import { CommonModule } from '@angular/common';
import { RouterLink, RouterLinkActive, RouterOutlet } from '@angular/router';

@Component({
  selector: 'swv-root',
  standalone: true,
  imports: [CommonModule, RouterOutlet, RouterLink, RouterLinkActive],
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  pageTitle = 'Star Wars Vehicle Sales';
  cartService = inject(CartService);

  // Computed signal for the cart count
  cartCount = signal(0);
}

```



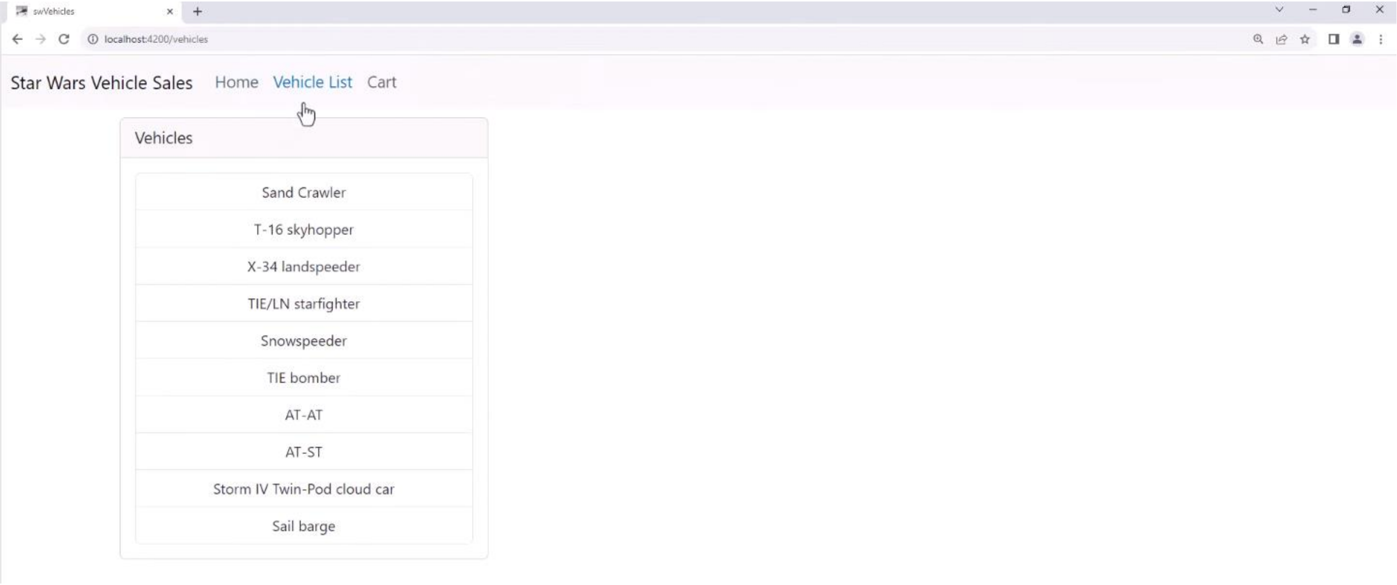
```
import { Component, computed, inject, signal } from '@angular/core';

import { CartService } from './cart/cart.service';
import { CommonModule } from '@angular/common';
import { RouterLink, RouterLinkActive, RouterOutlet } from '@angular/router';

@Component({
  selector: 'swv-root',
  standalone: true,
  imports: [CommonModule, RouterOutlet, RouterLink, RouterLinkActive],
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  pageTitle = 'Star Wars Vehicle Sales';
  cartService = inject(CartService);

  // Computed signal for the cart count
  cartCount = computed(() => this.cartService.cartItems().reduce(
    (acc, item) => acc + item.quantity, 0
  ));
}
```



Welcome to Star Wars Vehicle Sales

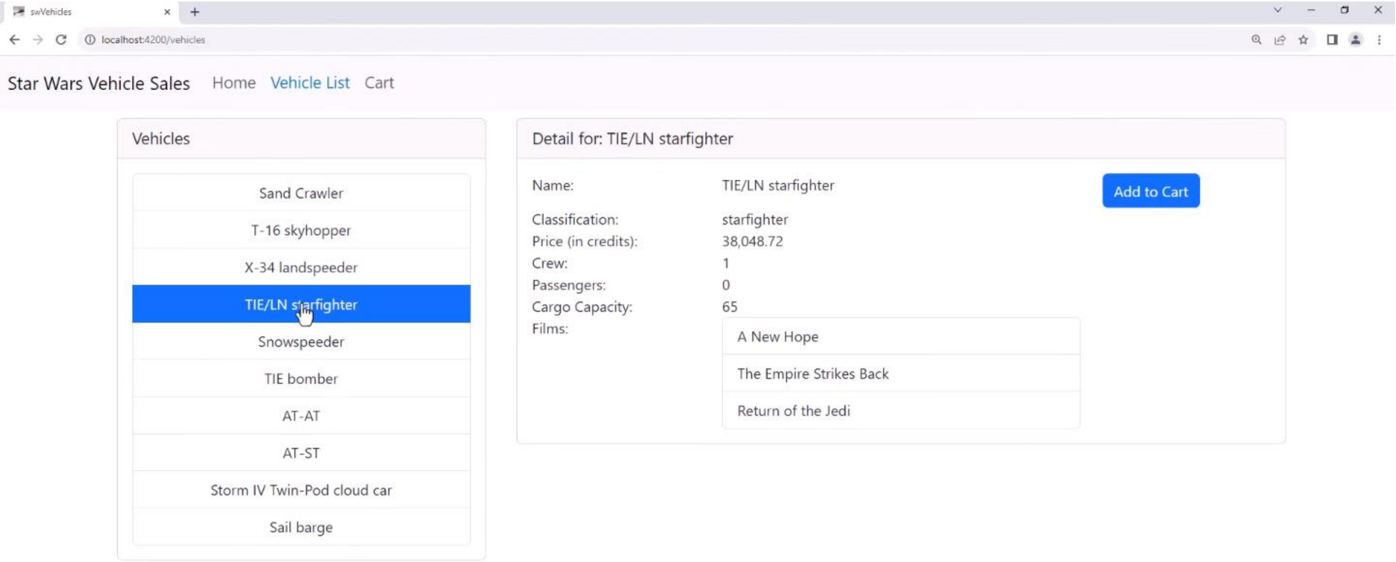
Developed by:

Deborah Kurata



Vehicles

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

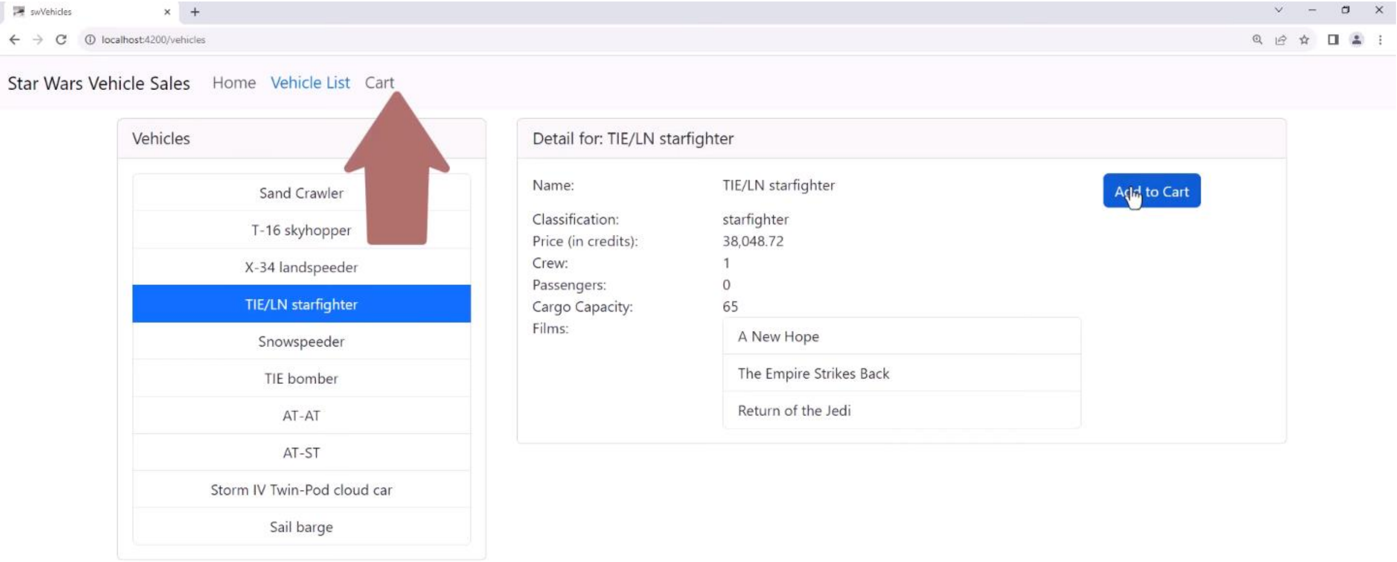


Vehicles

- 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

Detail for: TIE/LN starfighter

Name:	TIE/LN starfighter
Classification:	starfighter
Price (in credits):	38,048.72
Crew:	1
Passengers:	0
Cargo Capacity:	65
Films:	A New Hope The Empire Strikes Back Return of the Jedi

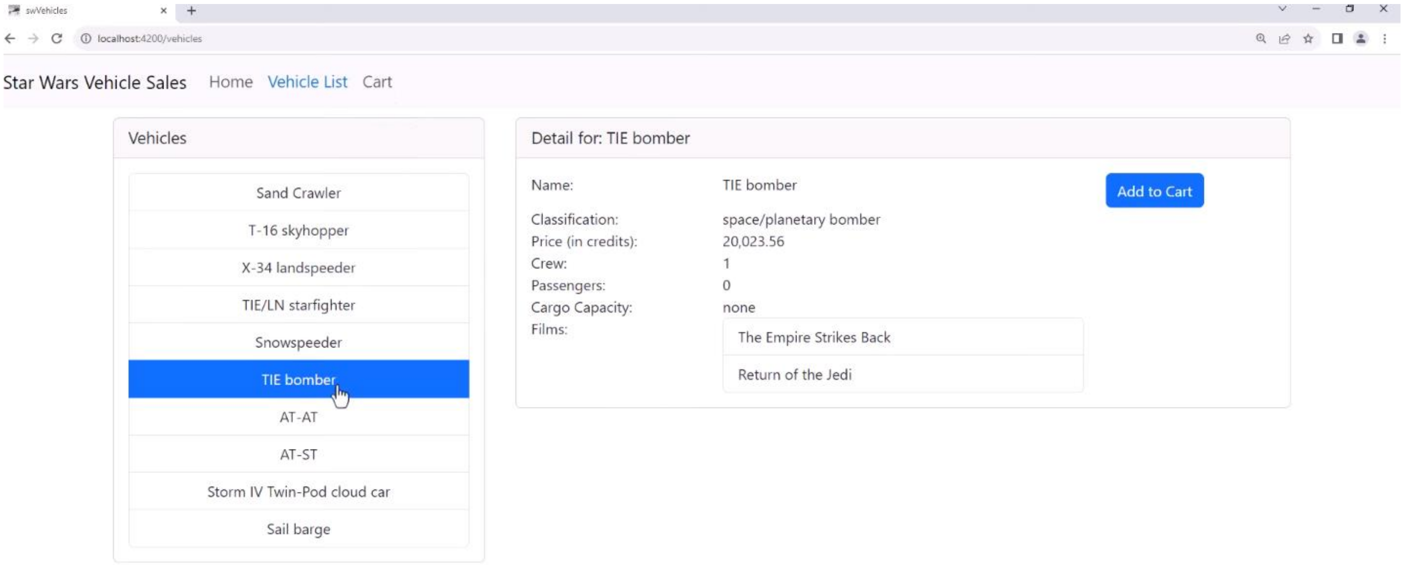


Vehicles

- 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

Detail for: TIE/LN starfighter

Name:	TIE/LN starfighter
Classification:	starfighter
Price (in credits):	38,048.72
Crew:	1
Passengers:	0
Cargo Capacity:	65
Films:	A New Hope The Empire Strikes Back Return of the Jedi

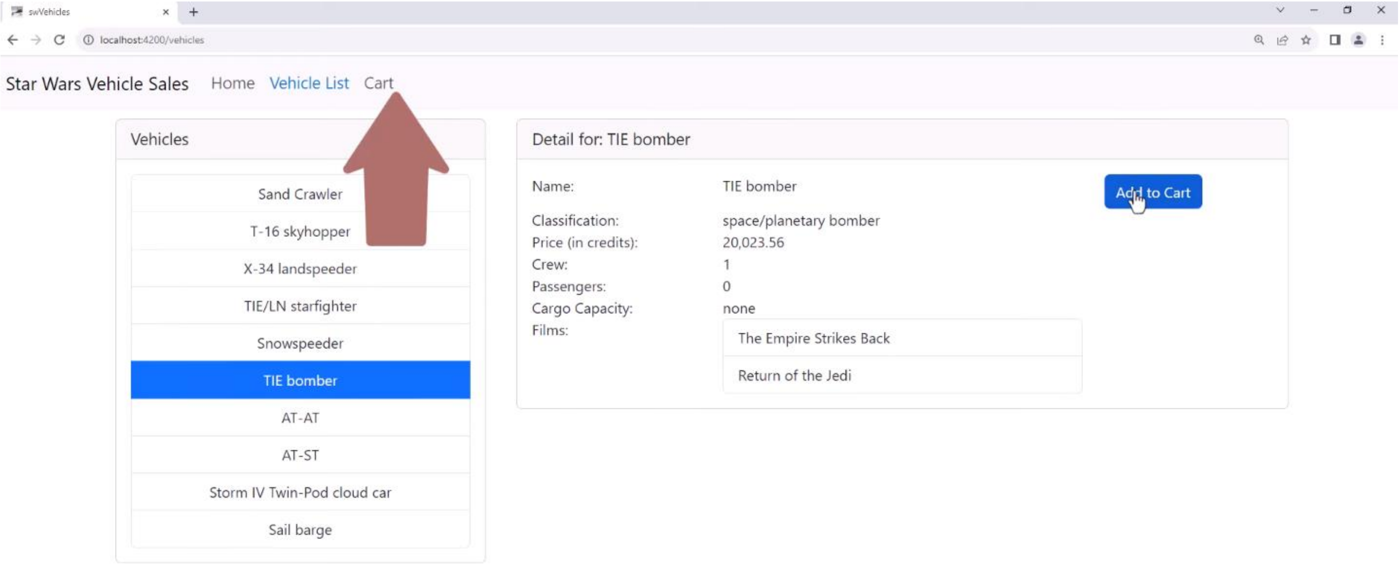


Vehicles

- 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

Detail for: TIE bomber

Name:	TIE bomber
Classification:	space/planetary bomber
Price (in credits):	20,023.56
Crew:	1
Passengers:	0
Cargo Capacity:	none
Films:	The Empire Strikes Back Return of the Jedi

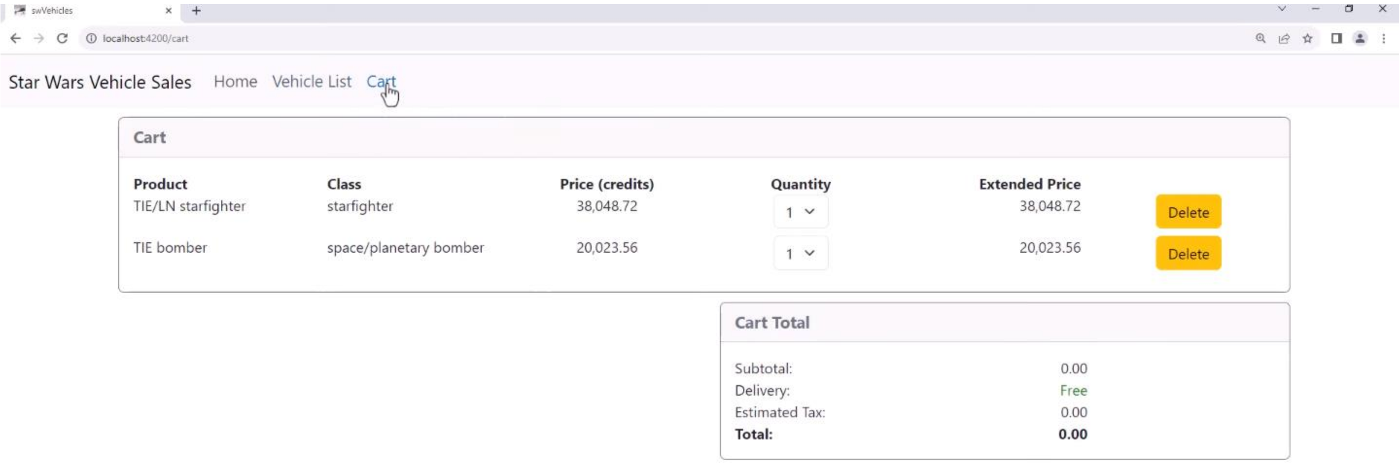


Vehicles

- 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

Detail for: TIE bomber

Name:	TIE bomber
Classification:	space/planetary bomber
Price (in credits):	20,023.56
Crew:	1
Passengers:	0
Cargo Capacity:	none
Films:	The Empire Strikes Back Return of the Jedi



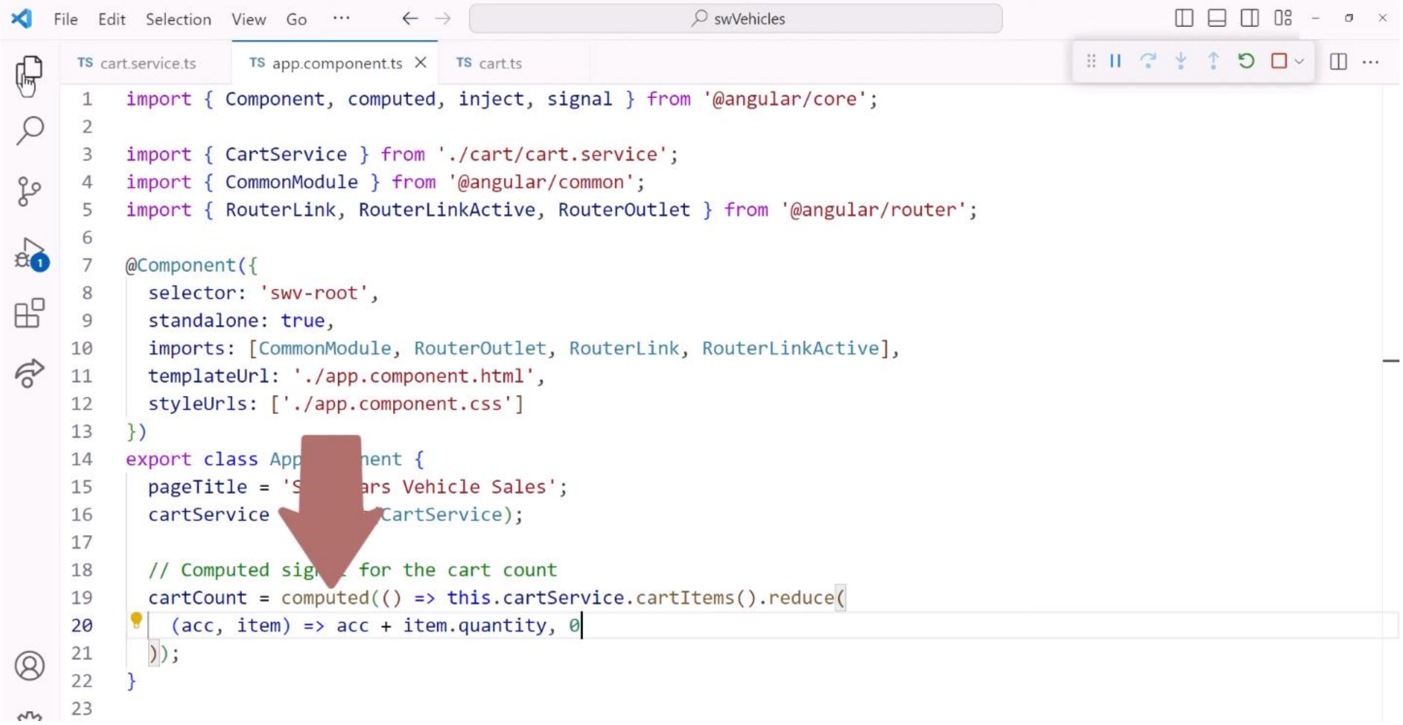
Cart

Product	Class	Price (credits)	Quantity	Extended Price	
TIE/LN starfighter	starfighter	38,048.72	1	38,048.72	Delete
TIE bomber	space/planetary bomber	20,023.56	1	20,023.56	Delete

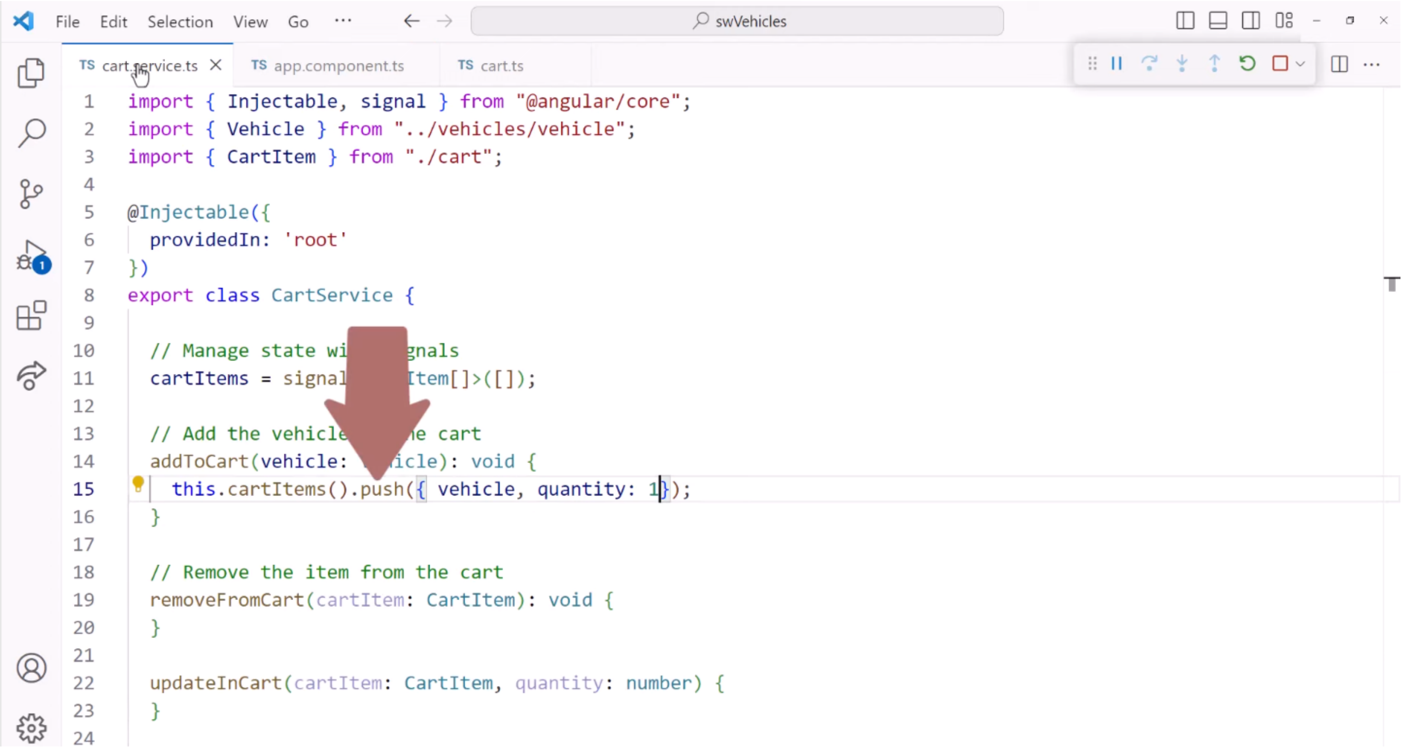
Cart Total

Subtotal:	0.00
Delivery:	Free
Estimated Tax:	0.00
Total:	**0.00**

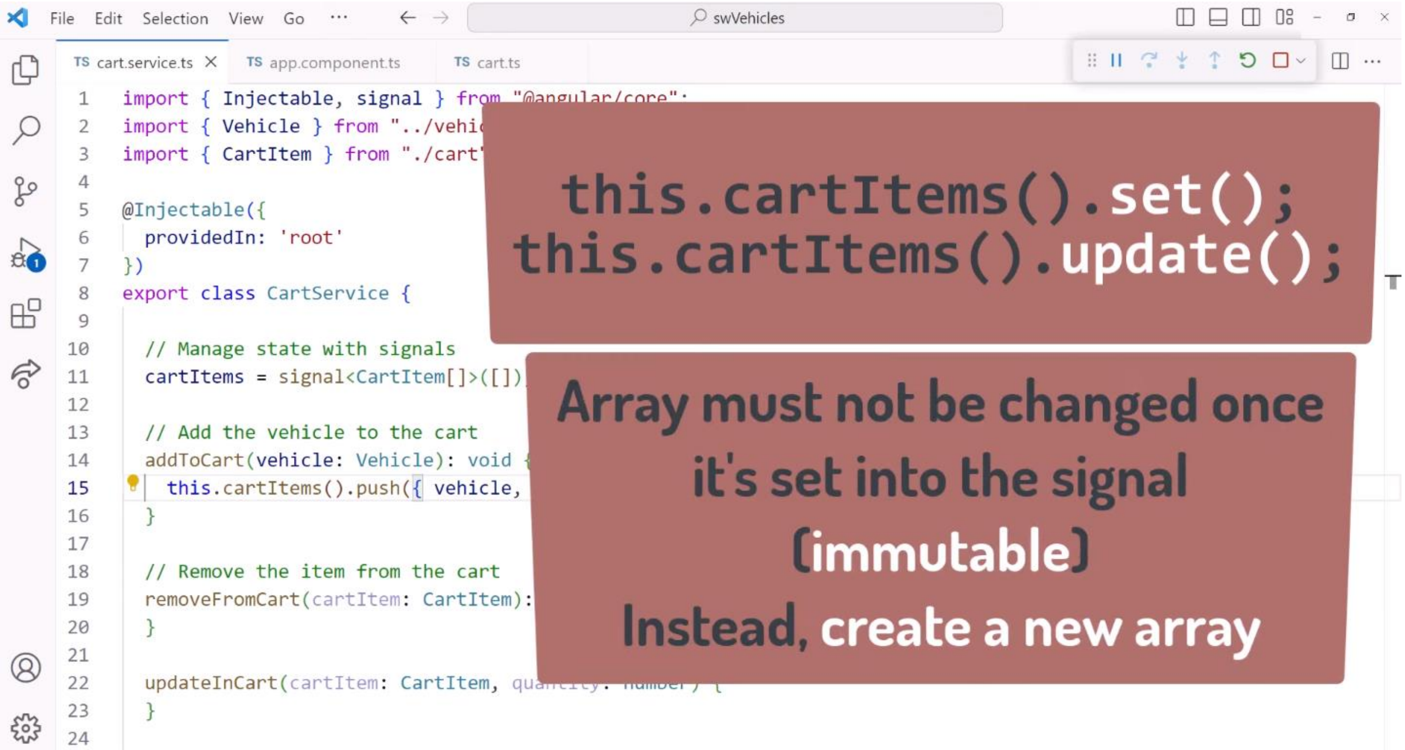
Our array is updating but our count badge does not display



```
import { Component, computed, inject, signal } from '@angular/core';

import { CartService } from './cart/cart.service';
import { CommonModule } from '@angular/common';
import { RouterLink, RouterLinkActive, RouterOutlet } from '@angular/router';

@Component({
  selector: 'swv-root',
  standalone: true,
  imports: [CommonModule, RouterOutlet, RouterLink, RouterLinkActive],
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class App     nent {
  pageTitle = 'S     ars Vehicle Sales';
  cartService          CartService);

  // Computed sig   for the cart count
  cartCount = computed(() => this.cartService.cartItems().reduce(
    (acc, item) => acc + item.quantity, 0
  ));
}
```



```
import { Injectable, signal } from "@angular/core";
import { Vehicle } from "../vehicles/vehicle";
import { CartItem } from "./cart";

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

  // Manage state with signals
  cartItems = signal<CartItem[]>([]);

  // Add the vehicle to the cart
  addToCart(vehicle: Vehicle): void {
    this.cartItems().push({ vehicle, quantity: 1});
  }

  // Remove the item from the cart
  removeFromCart(cartItem: CartItem): void {
  }

  updateInCart(cartItem: CartItem, quantity: number) {
  }
```

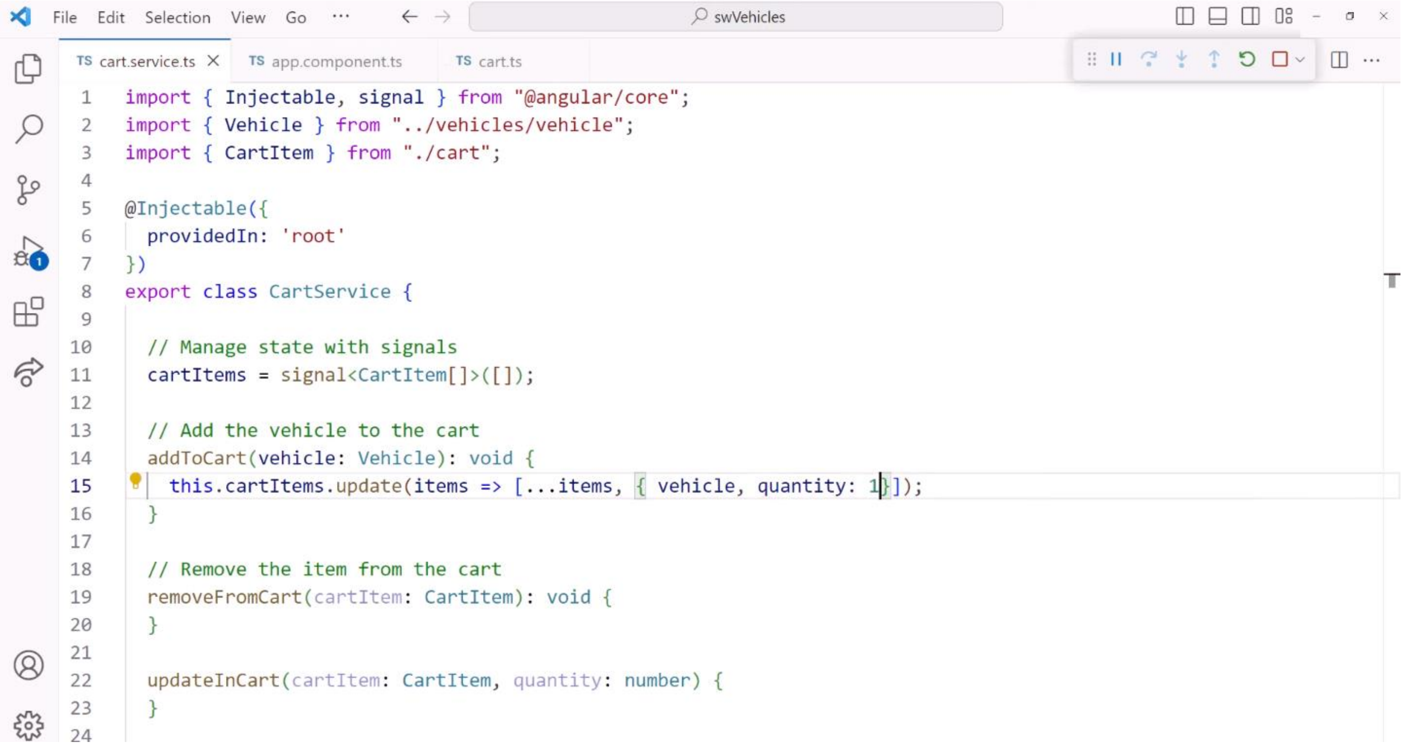


```
import { Injectable, signal } from "@angular/core";
import { Vehicle } from "../vehic
import { CartItem } from "./cart"

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

  // Manage state with signals
  cartItems = signal<CartItem[]>([])

  // Add the vehicle to the cart
  addToCart(vehicle: Vehicle): void {
    this.cartItems().push({ vehicle,
  }

  // Remove the item from the cart
  removeFromCart(cartItem: CartItem):
  }

  updateInCart(cartItem: CartItem, quantity: number) {
  }
```

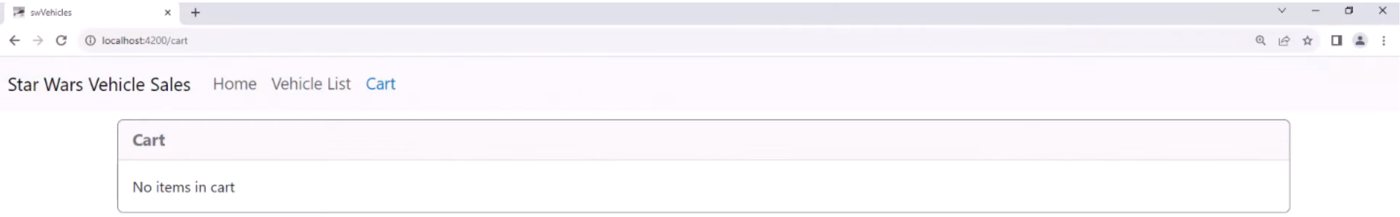
```
this.cartItems().set();
this.cartItems().update();
```

Array must not be changed once it's set into the signal (immutable)

Instead, create a new array

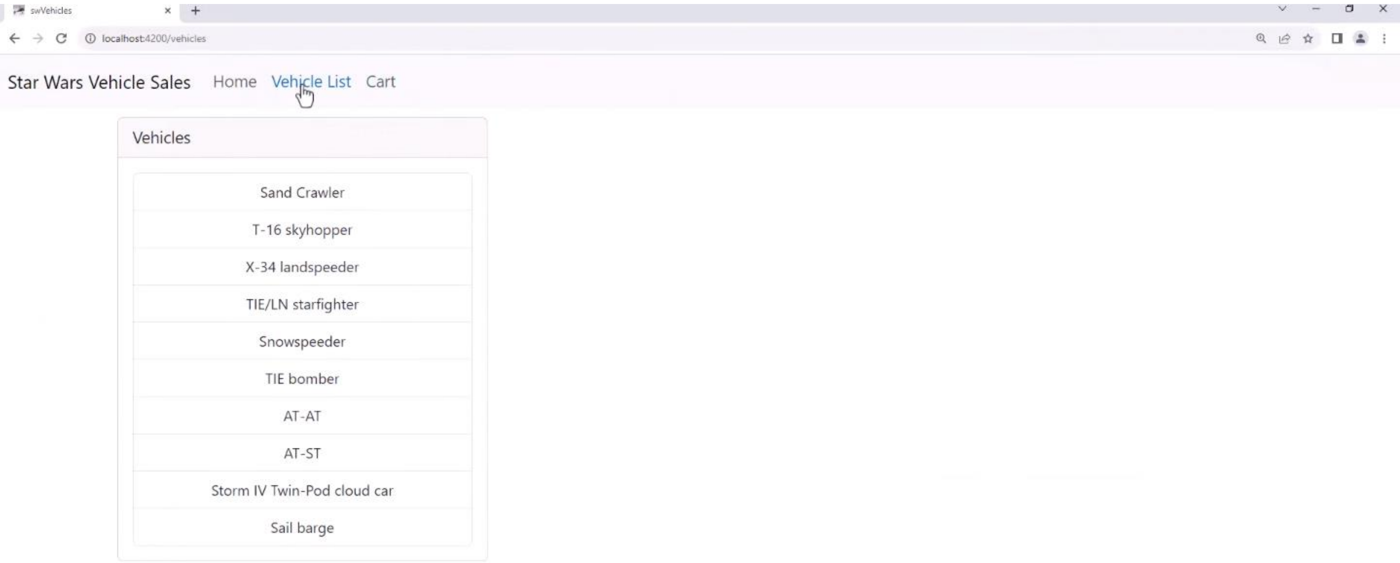


```
import { Injectable, signal } from "@angular/core";
import { Vehicle } from "../vehicles/vehicle";
import { CartItem } from "./cart";

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

  // Manage state with signals
  cartItems = signal<CartItem[]>([]);

  // Add the vehicle to the cart
  addToCart(vehicle: Vehicle): void {
    this.cartItems.update(items => [...items, { vehicle, quantity: 1}]);
  }

  // Remove the item from the cart
  removeFromCart(cartItem: CartItem): void {
  }

  updateInCart(cartItem: CartItem, quantity: number) {
  }

```



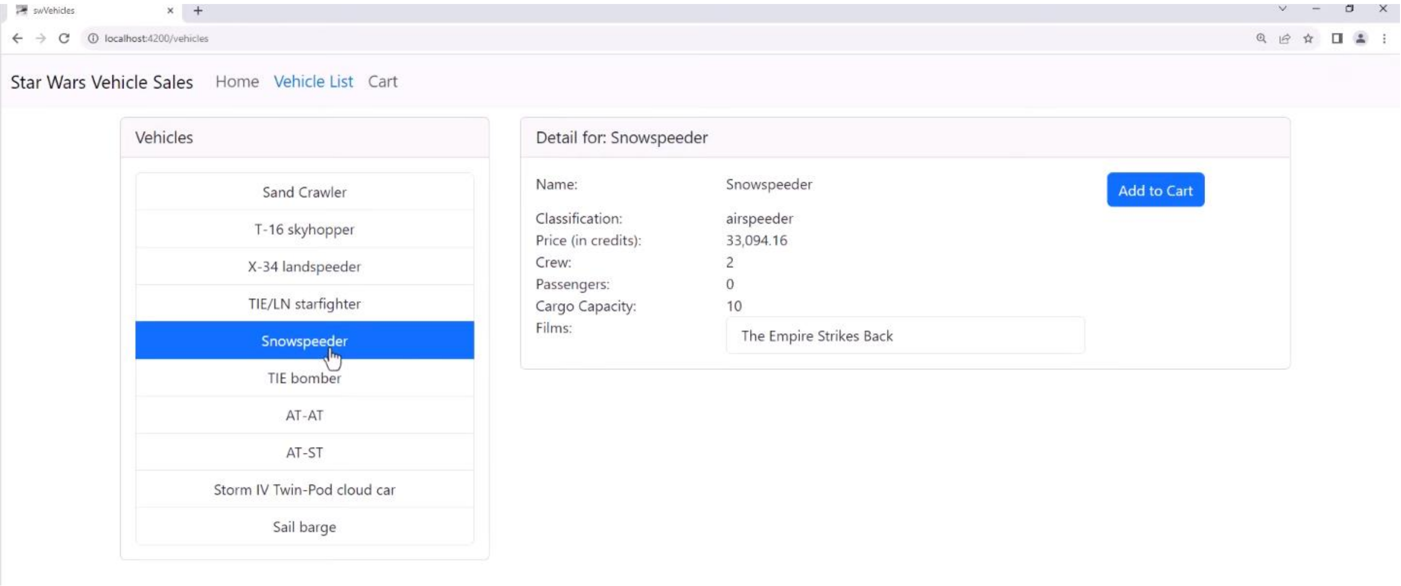
Cart

No items in cart



Vehicles

- 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

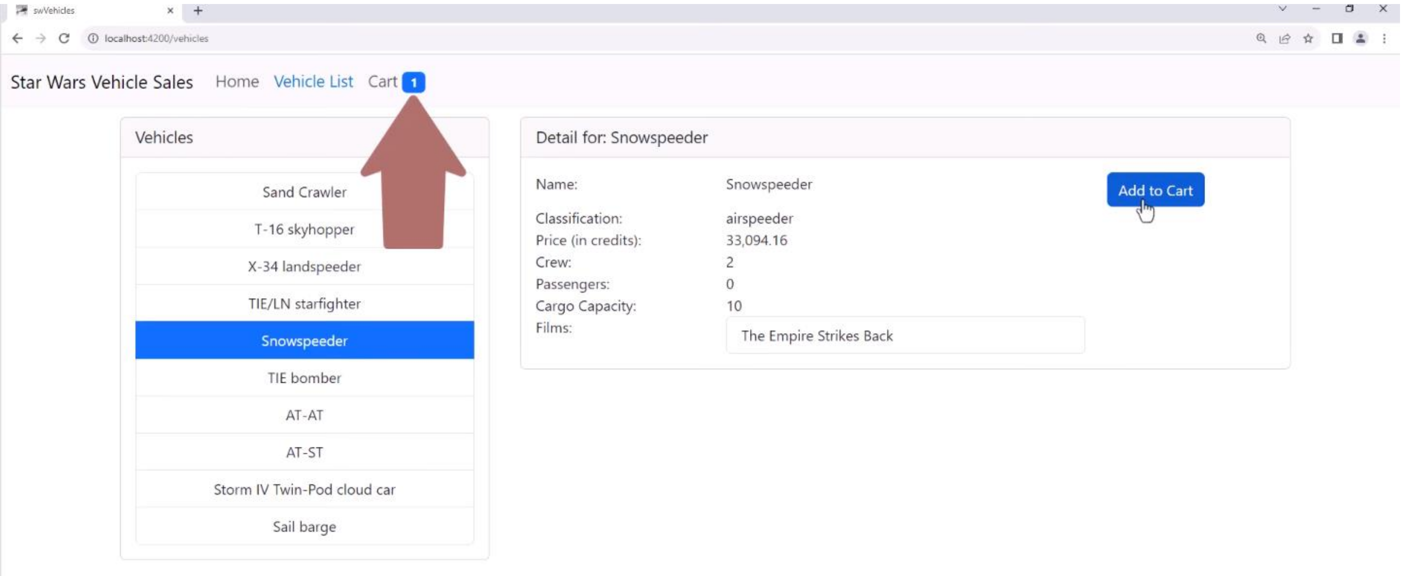


Vehicles

- 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

Detail for: Snowspeeder

Name:	Snowspeeder
Classification:	airspeeder
Price (in credits):	33,094.16
Crew:	2
Passengers:	0
Cargo Capacity:	10
Films:	The Empire Strikes Back



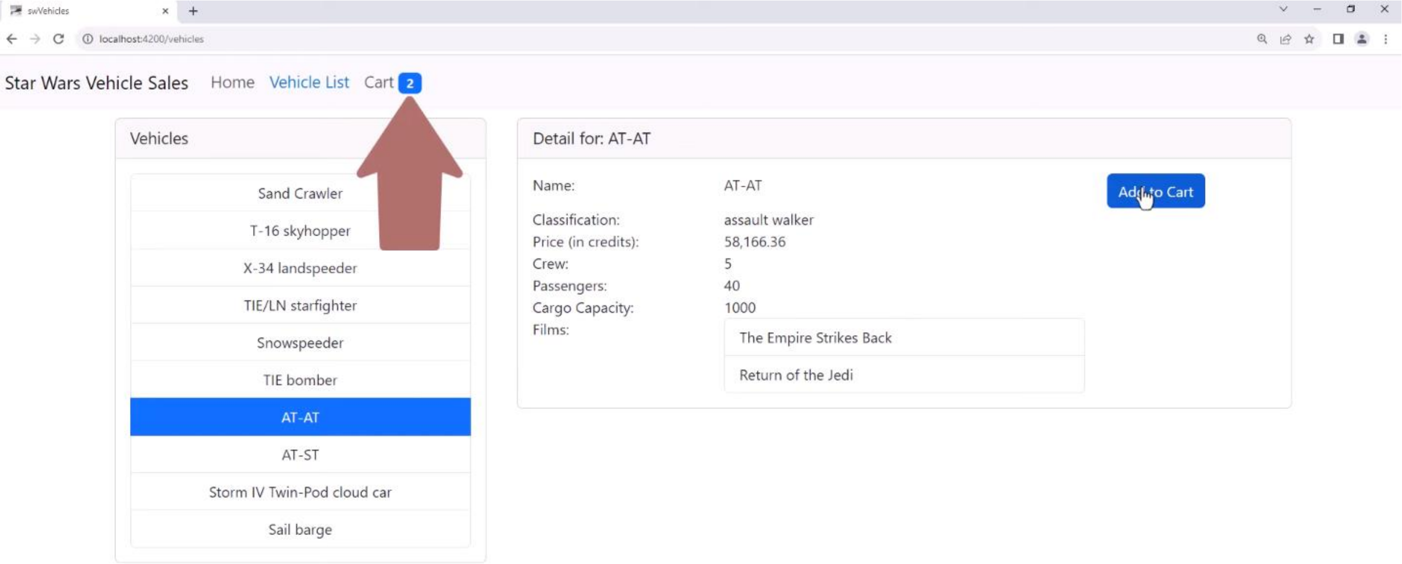
Vehicles

- 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

Detail for: Snowspeeder

Name:	Snowspeeder
Classification:	airspeeder
Price (in credits):	33,094.16
Crew:	2
Passengers:	0
Cargo Capacity:	10
Films:	The Empire Strikes Back

Our computed() signal reacted to the cartItem signal change and recalculated itself

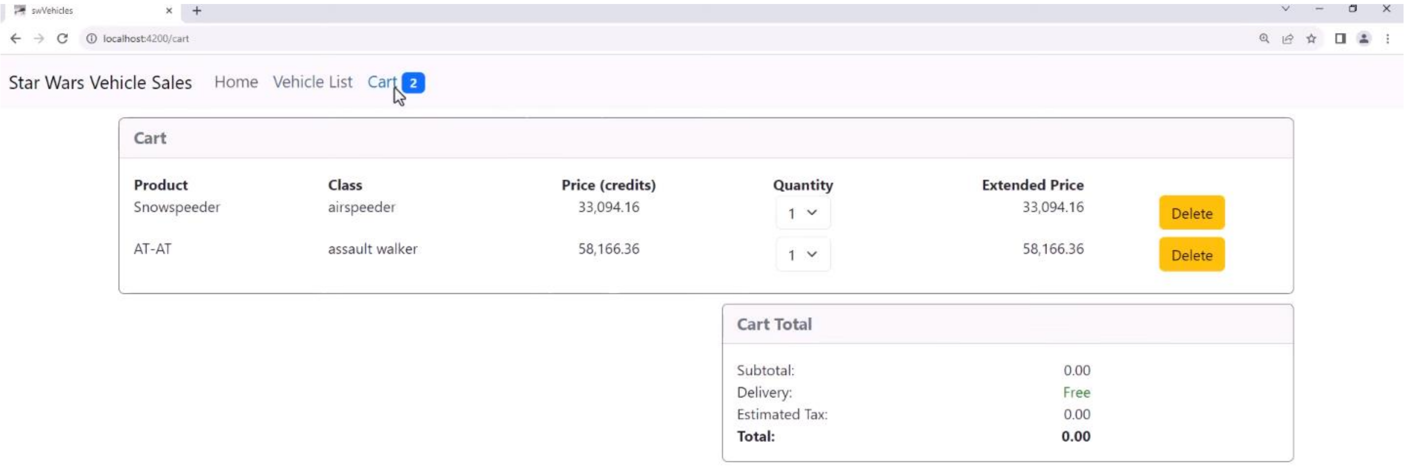


Vehicles

- 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

Detail for: AT-AT

Name:	AT-AT
Classification:	assault walker
Price (in credits):	58,166.36
Crew:	5
Passengers:	40
Cargo Capacity:	1000
Films:	The Empire Strikes Back Return of the Jedi



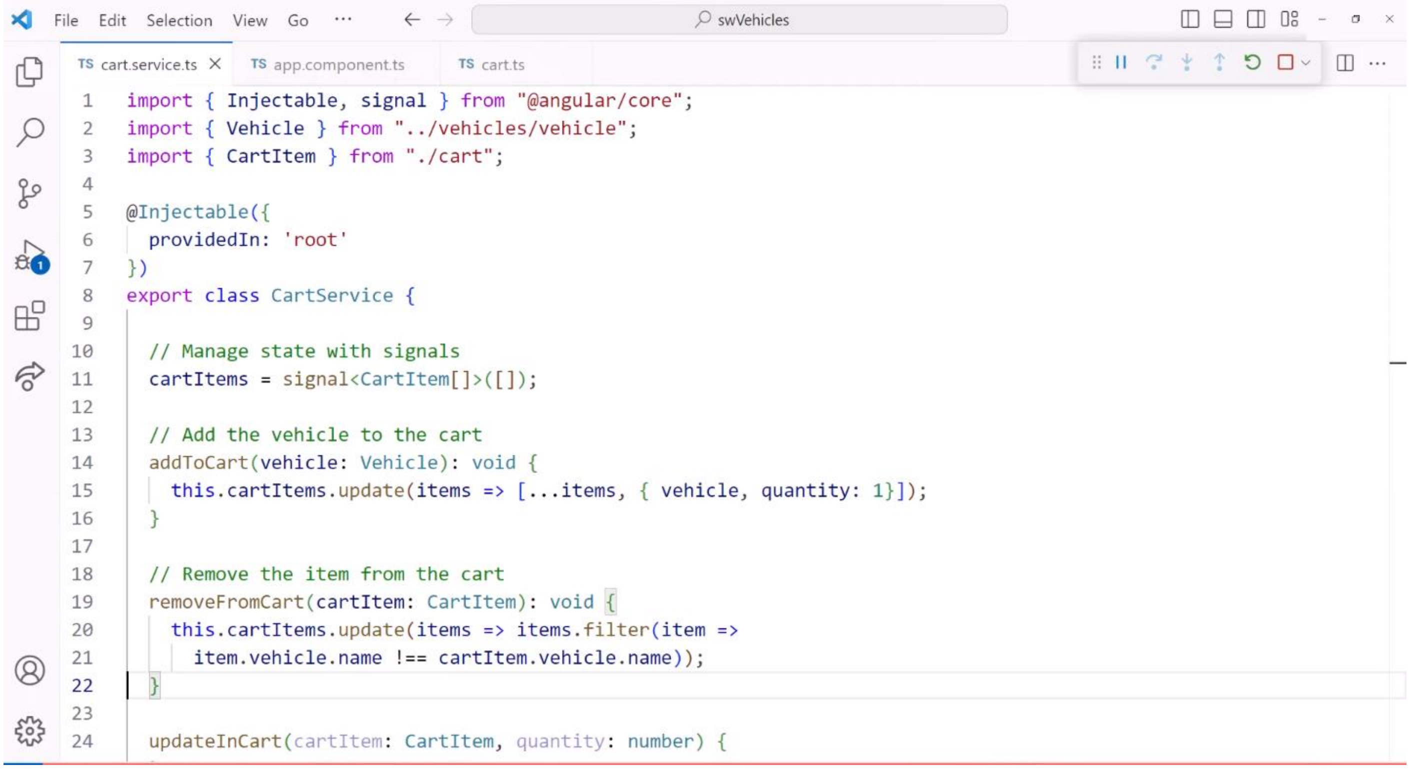
Star Wars Vehicle Sales Home Vehicle List Cart 2

Cart

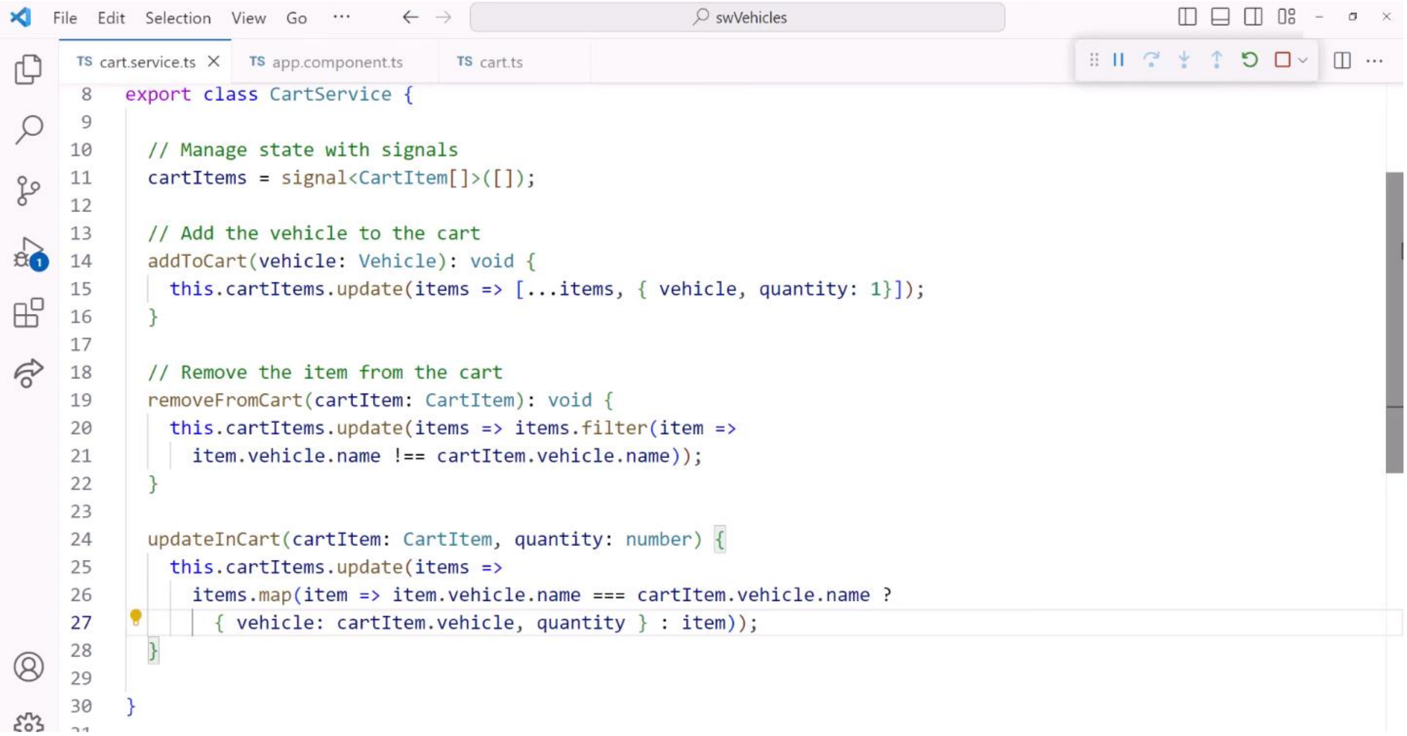
Product	Class	Price (credits)	Quantity	Extended Price	
Snowspeeder	airspeeder	33,094.16	1	33,094.16	Delete
AT-AT	assault walker	58,166.36	1	58,166.36	Delete

Cart Total

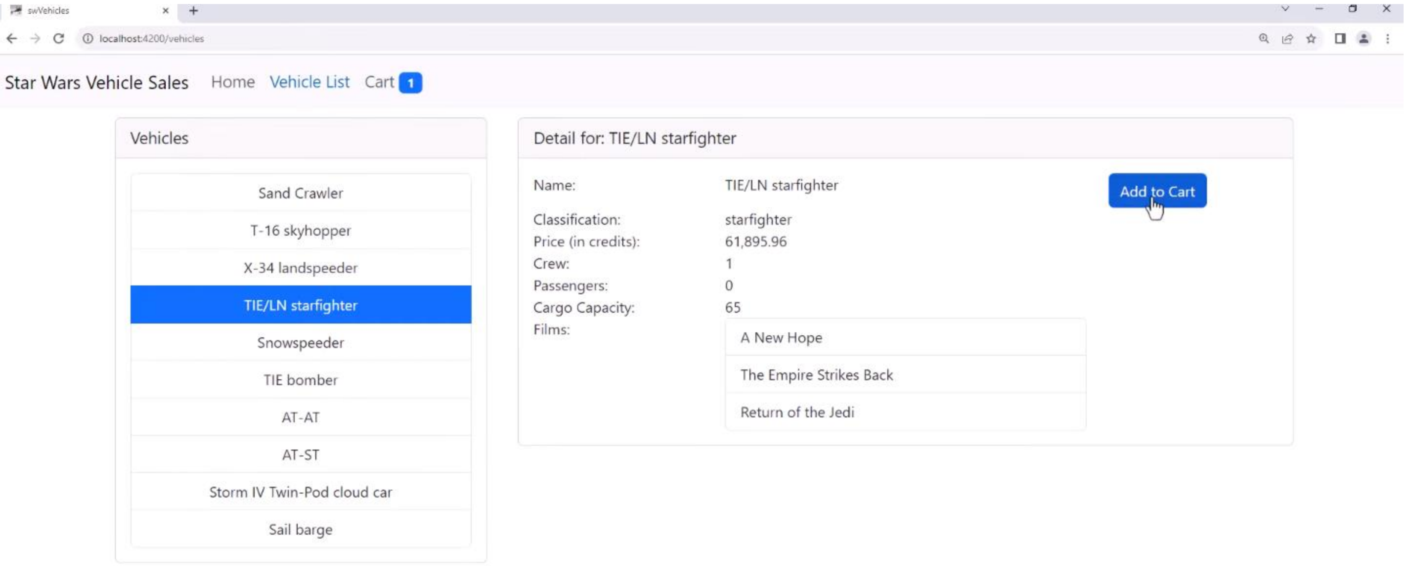
Subtotal:	0.00
Delivery:	Free
Estimated Tax:	0.00
Total:	**0.00**



```
import { Injectable, signal } from "@angular/core";
import { Vehicle } from "../vehicles/vehicle";
import { CartItem } from "./cart";

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

  // Manage state with signals
  cartItems = signal<CartItem[]>([]);

  // Add the vehicle to the cart
  addToCart(vehicle: Vehicle): void {
    this.cartItems.update(items => [...items, { vehicle, quantity: 1}]);
  }

  // Remove the item from the cart
  removeFromCart(cartItem: CartItem): void {
    this.cartItems.update(items => items.filter(item =>
      item.vehicle.name !== cartItem.vehicle.name));
  }

  updateInCart(cartItem: CartItem, quantity: number) {
```



```
export class CartService {

  // Manage state with signals
  cartItems = signal<CartItem[]>([]);

  // Add the vehicle to the cart
  addToCart(vehicle: Vehicle): void {
    this.cartItems.update(items => [...items, { vehicle, quantity: 1}]);
  }

  // Remove the item from the cart
  removeFromCart(cartItem: CartItem): void {
    this.cartItems.update(items => items.filter(item =>
      item.vehicle.name !== cartItem.vehicle.name));
  }

  updateInCart(cartItem: CartItem, quantity: number) {
    this.cartItems.update(items =>
      items.map(item => item.vehicle.name === cartItem.vehicle.name ?
        { vehicle: cartItem.vehicle, quantity } : item));
  }

}
```

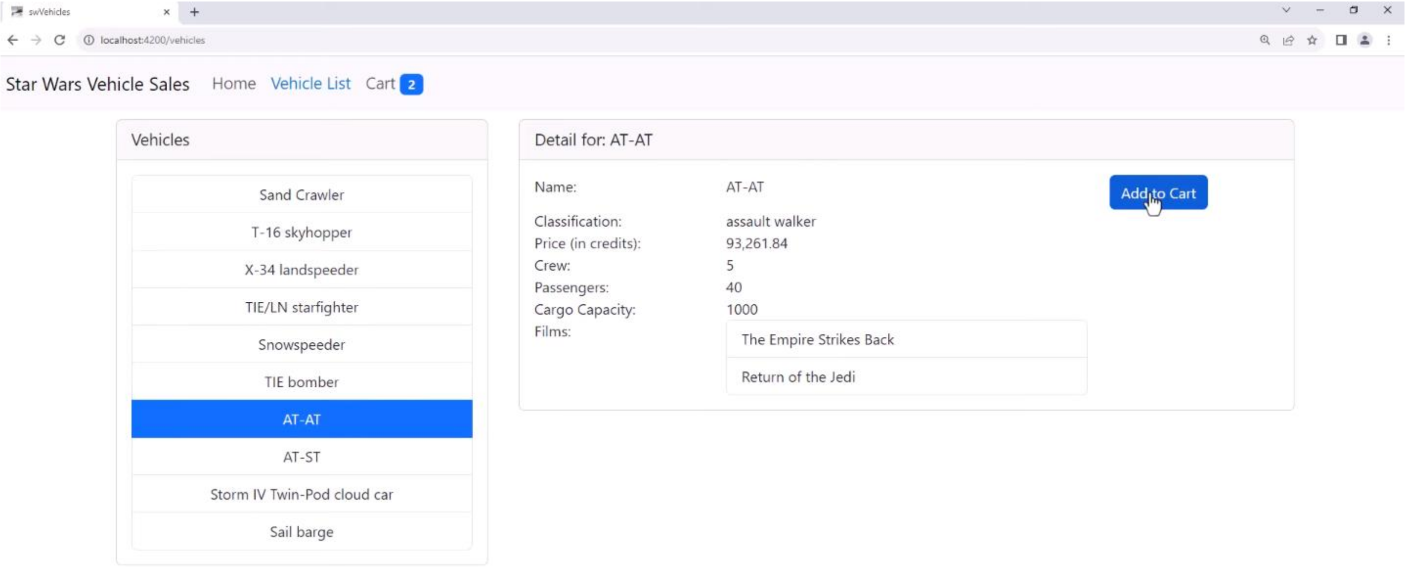


Vehicles

- 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

Detail for: TIE/LN starfighter

Name:	TIE/LN starfighter
Classification:	starfighter
Price (in credits):	61,895.96
Crew:	1
Passengers:	0
Cargo Capacity:	65
Films:	A New Hope The Empire Strikes Back Return of the Jedi

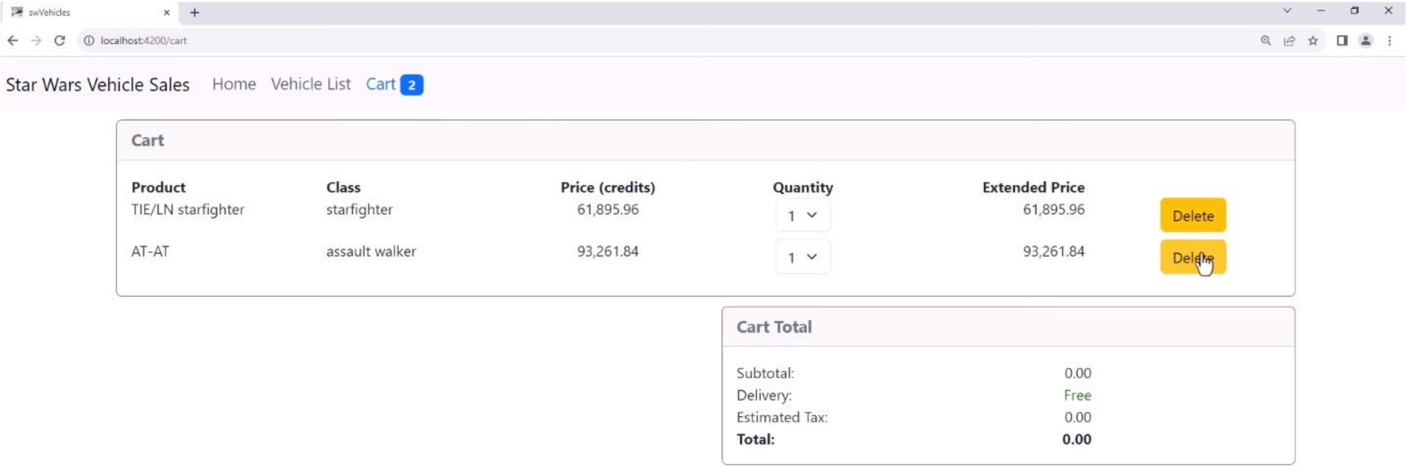


Vehicles

- 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

Detail for: AT-AT

Name:	AT-AT
Classification:	assault walker
Price (in credits):	93,261.84
Crew:	5
Passengers:	40
Cargo Capacity:	1000
Films:	The Empire Strikes Back Return of the Jedi

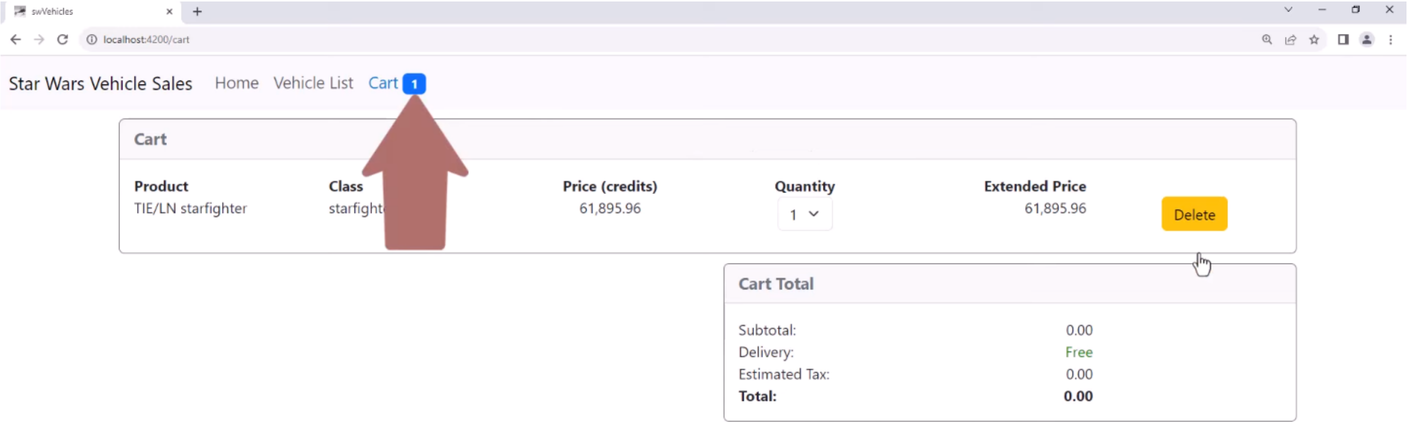


Cart

Product	Class	Price (credits)	Quantity	Extended Price	
TIE/LN starfighter	starfighter	61,895.96	1	61,895.96	Delete
AT-AT	assault walker	93,261.84	1	93,261.84	Delete

Cart Total

Subtotal:	0.00
Delivery:	Free
Estimated Tax:	0.00
Total:	**0.00**

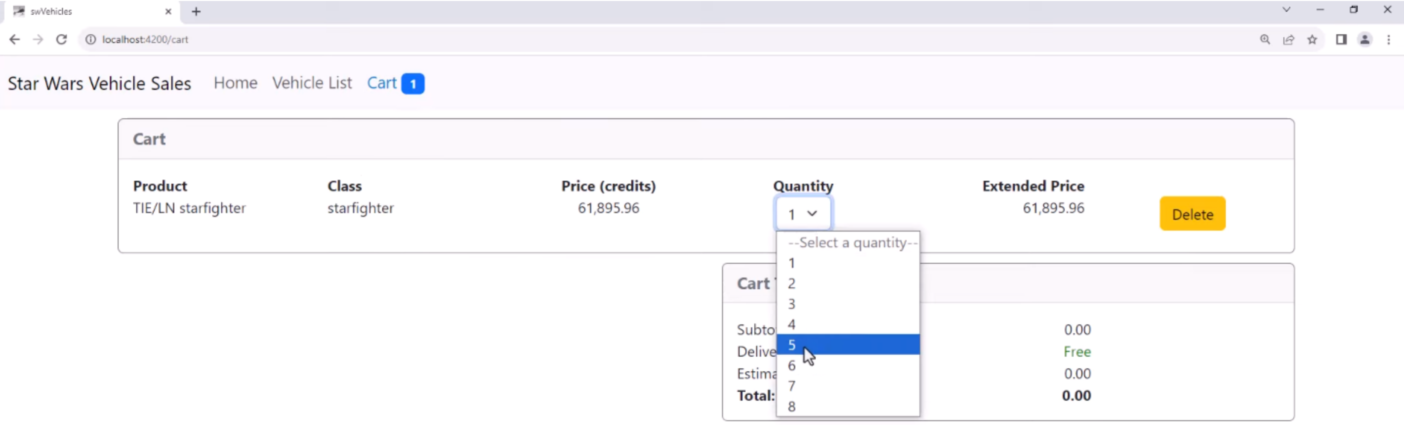


Cart

Product	Class	Price (credits)	Quantity	Extended Price	
TIE/LN starfighter	starfight	61,895.96	1	61,895.96	Delete

Cart Total

Subtotal:	0.00
Delivery:	Free
Estimated Tax:	0.00
Total:	**0.00**

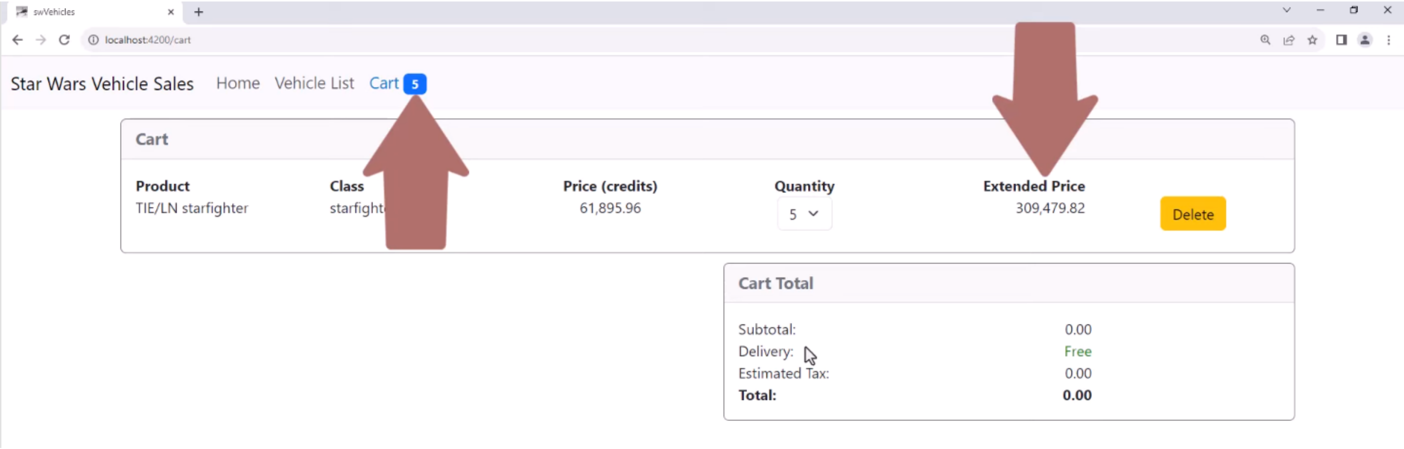


Cart

Product	Class	Price (credits)	Quantity	Extended Price	
TIE/LN starfighter	starfighter	61,895.96	1	61,895.96	Delete

Cart T

Subto	0.00
Delive	Free
Estima	0.00
Total:	**0.00**

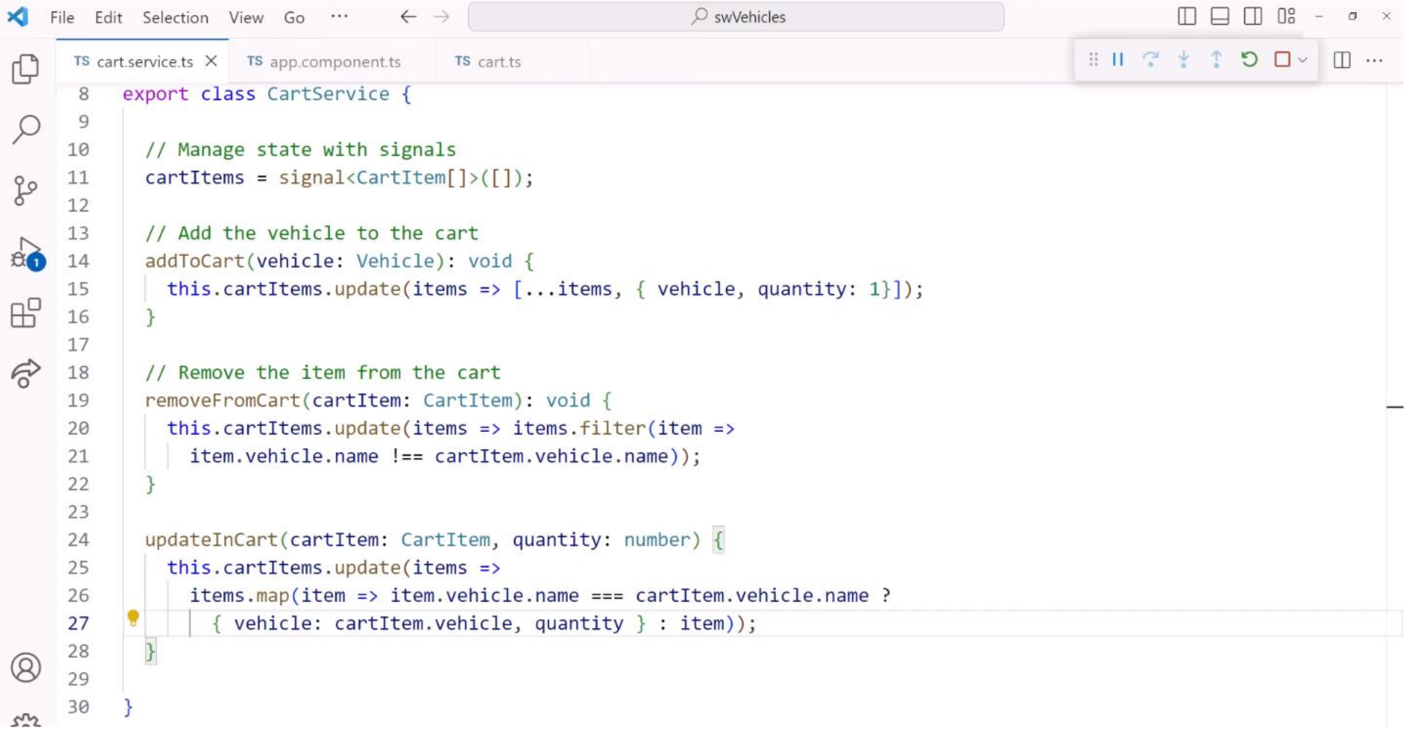


Cart

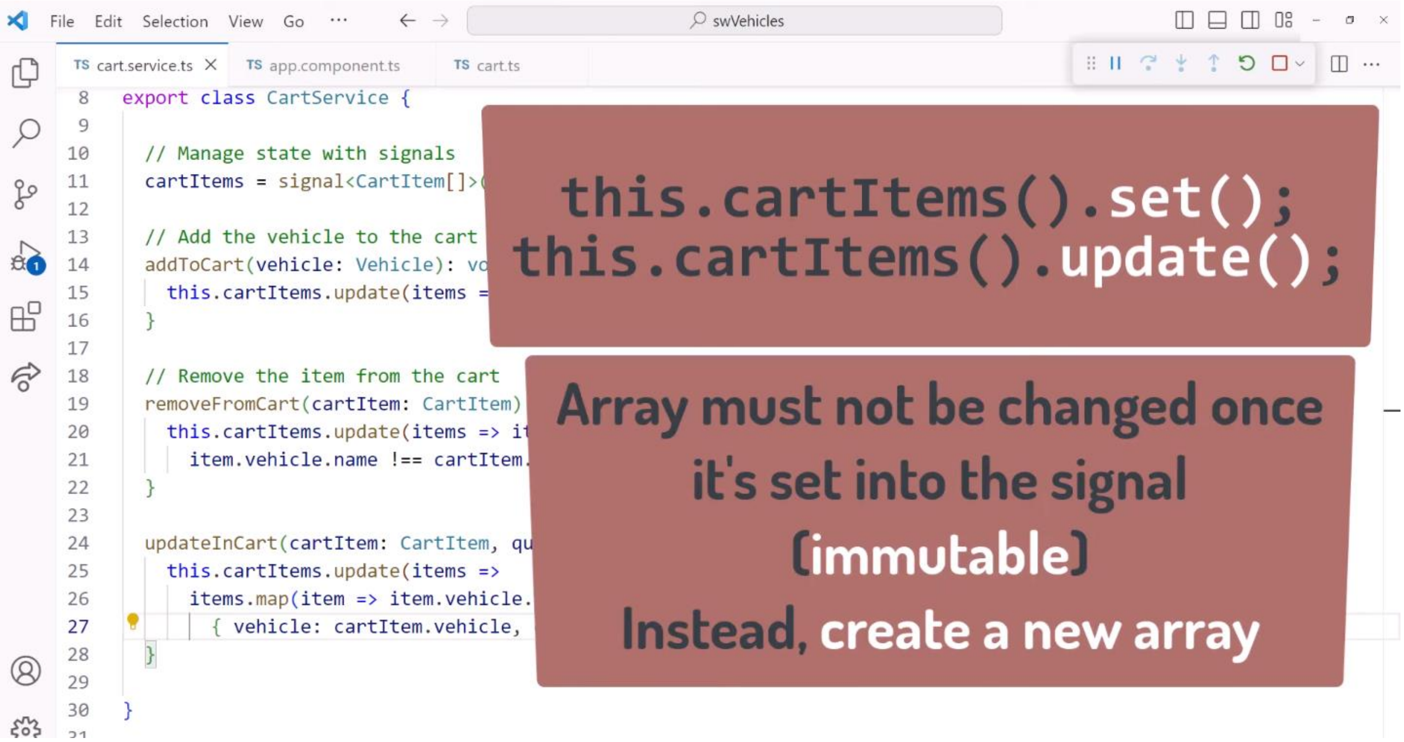
Product	Class	Price (credits)	Quantity	Extended Price	
TIE/LN starfighter	starfight	61,895.96	5	309,479.82	Delete

Cart Total

Subtotal:	0.00
Delivery:	Free
Estimated Tax:	0.00
Total:	**0.00**



```
export class CartService {

  // Manage state with signals
  cartItems = signal<CartItem[]>([]);

  // Add the vehicle to the cart
  addToCart(vehicle: Vehicle): void {
    this.cartItems.update(items => [...items, { vehicle, quantity: 1}]);
  }

  // Remove the item from the cart
  removeFromCart(cartItem: CartItem): void {
    this.cartItems.update(items => items.filter(item =>
      item.vehicle.name !== cartItem.vehicle.name));
  }

  updateInCart(cartItem: CartItem, quantity: number) {
    this.cartItems.update(items =>
      items.map(item => item.vehicle.name === cartItem.vehicle.name ?
        { vehicle: cartItem.vehicle, quantity } : item));
  }

}
```



```
export class CartService {

  // Manage state with signals
  cartItems = signal<CartItem[]>(

  // Add the vehicle to the cart
  addToCart(vehicle: Vehicle): vo
    this.cartItems.update(items =
  }

  // Remove the item from the cart
  removeFromCart(cartItem: CartItem)
    this.cartItems.update(items => it
      item.vehicle.name !== cartItem.
  }

  updateInCart(cartItem: CartItem, qu
    this.cartItems.update(items =>
      items.map(item => item.vehicle.
        { vehicle: cartItem.vehicle,
  }

}
```

```
this.cartItems().set();
this.cartItems().update();
```

Array must not be changed once it's set into the signal (immutable)

Instead, create a new array