

Signals in Action: Building an App

Deborah Kurata
19.5K subscribers

Subscribe

365



Share

Download

Thanks

...

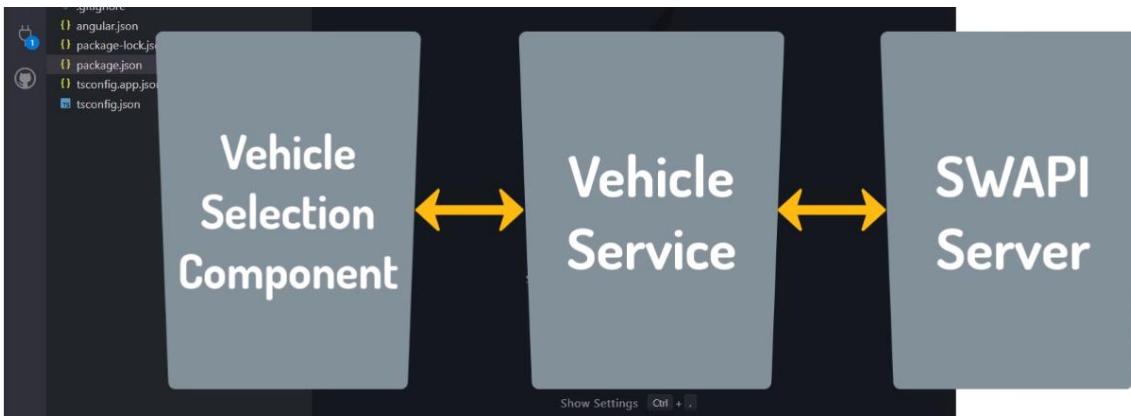
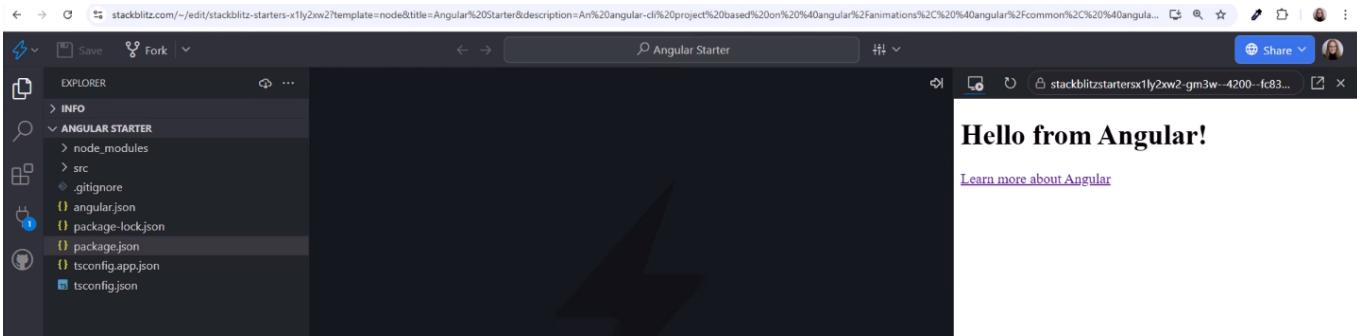
5,906 views Dec 11, 2024 #angular #resource #angularsignals

Learning signals is only the first step. Often, the more difficult next step is understanding when and how to apply signals in an application.

In this video, we build a simple application using many of the new signal features. We start by creating a signal-based service. This service uses SWAPI (the public Star Wars API) to retrieve data and expose that data as signals. Then we build a simple component to use those signals.

00:00 Applying signals in an application
00:38 Sample application
01:26 Creating a new application
01:53 Vehicle service
03:55 Vehicle signals
04:46 Vehicle computed signals

06:04 Vehicle selection component
08:15 Displaying our component
09:00 Retrieving data as a signal
11:26 Confirming our results
11:57 Resetting a signal with linkedSignal()
13:22 Benefits of signals



```
1 import { provideHttpClient } from '@angular/common/http';
2 import { Component } from '@angular/core';
3 import { bootstrapApplication } from '@angular/platform-browser';
4 import { VehicleSelectionComponent } from './app/vehicle-selection/vehicle-selection.component';
5
6 @Component({
7   selector: 'app-root',
8   imports: [VehicleSelectionComponent],
9   template: `
10     <app-vehicle-selection>
11   `,
12 })
13 export class App {
14   name = 'Angular';
15 }
16
17 bootstrapApplication(App, {
18   providers: [provideHttpClient()]
19 });
```

StackBlitz editor showing the main.ts file:

```
1 import { provideHttpClient } from '@angular/common/http';
2 import { Component } from '@angular/core';
3 import { bootstrapApplication } from '@angular/platform-browser';
4 import { VehicleSelectionComponent } from './app/vehicle-selection/vehicle-selection.component';
5
6 @Component({
7   selector: 'app-root',
8   imports: [ VehicleSelectionComponent ],
9   template: `
10     <app-vehicle-selection/>
11   `,
12 })
13 export class App {
14   name = 'Angular';
15 }
16
17 bootstrapApplication(App, {
18   providers: [ provideHttpClient() ]
19 });
20
```

The right side shows a modal titled "Star Wars Vehicle Sales" with a dropdown menu:

- 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

StackBlitz editor showing the main.ts file:

```
1 import { provideHttpClient } from '@angular/common/http';
2 import { Component } from '@angular/core';
3 import { bootstrapApplication } from '@angular/platform-browser';
4 import { VehicleSelectionComponent } from './app/vehicle-selection/vehicle-selection.component';
5
6 @Component({
7   selector: 'app-root',
8   imports: [ VehicleSelectionComponent ],
9   template: `
10     <app-vehicle-selection/>
11   `,
12 })
13 export class App {
14   name = 'Angular';
15 }
16
17 bootstrapApplication(App, {
18   providers: [ provideHttpClient() ]
19 });
20
```

The right side shows the same modal, but the dropdown has been populated with "X-34 landspeeder". A yellow arrow points from the dropdown to the modal.

Modal content:

- X-34 landspeeder
- Quantity: 1
- Vehicle: X-34 landspeeder
- Price: 10550
- Total: 10550

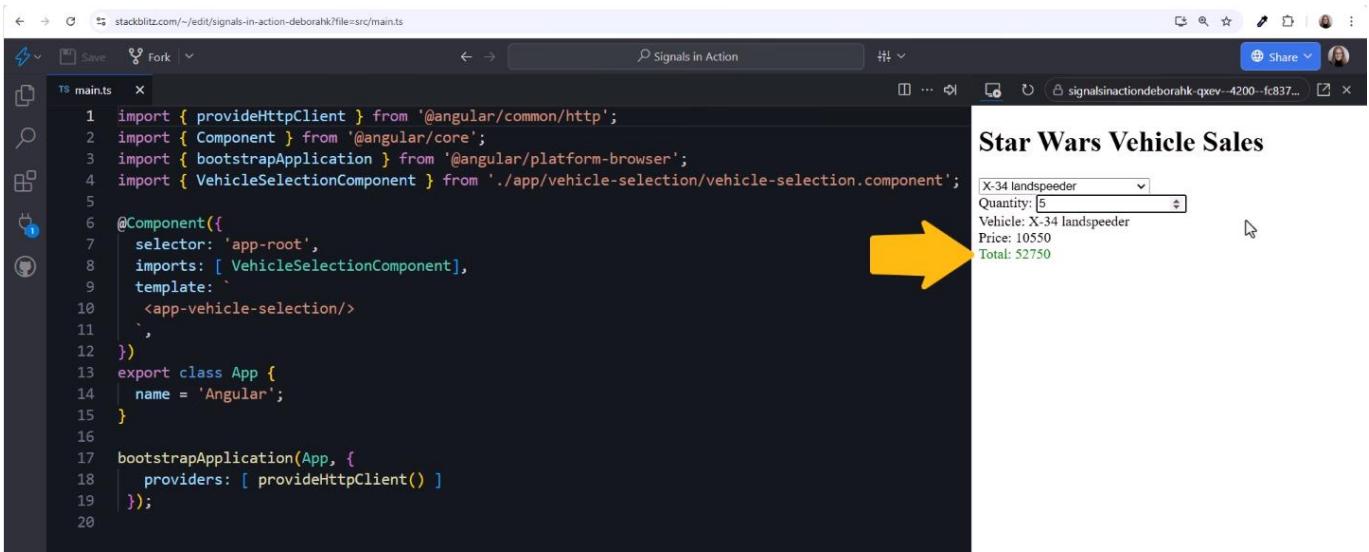
StackBlitz editor showing the main.ts file:

```
1 import { provideHttpClient } from '@angular/common/http';
2 import { Component } from '@angular/core';
3 import { bootstrapApplication } from '@angular/platform-browser';
4 import { VehicleSelectionComponent } from './app/vehicle-selection/vehicle-selection.component';
5
6 @Component({
7   selector: 'app-root',
8   imports: [ VehicleSelectionComponent ],
9   template: `
10     <app-vehicle-selection/>
11   `,
12 })
13 export class App {
14   name = 'Angular';
15 }
16
17 bootstrapApplication(App, {
18   providers: [ provideHttpClient() ]
19 });
20
```

The right side shows the same modal, but the dropdown now contains "X-34 landspeeder" with a quantity of 3. A yellow arrow points from the dropdown to the modal.

Modal content:

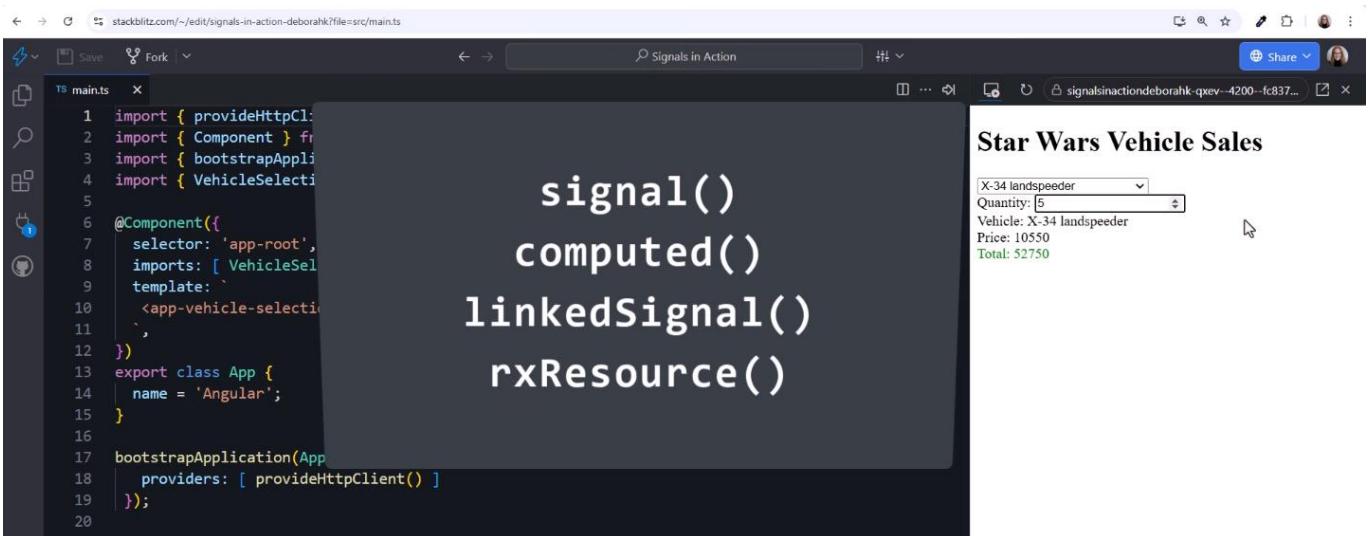
- X-34 landspeeder
- Quantity: 3
- Vehicle: X-34 landspeeder
- Price: 10550
- Total: 31650



```
TS main.ts
1 import { provideHttpClient } from '@angular/common/http';
2 import { Component } from '@angular/core';
3 import { bootstrapApplication } from '@angular/platform-browser';
4 import { VehicleSelectionComponent } from './app/vehicle-selection/vehicle-selection.component';
5
6 @Component({
7   selector: 'app-root',
8   imports: [ VehicleSelectionComponent ],
9   template: `
10     <app-vehicle-selection>
11   `,
12 })
13 export class App {
14   name = 'Angular';
15 }
16
17 bootstrapApplication(App, {
18   providers: [ provideHttpClient() ]
19 });
20
```

Star Wars Vehicle Sales

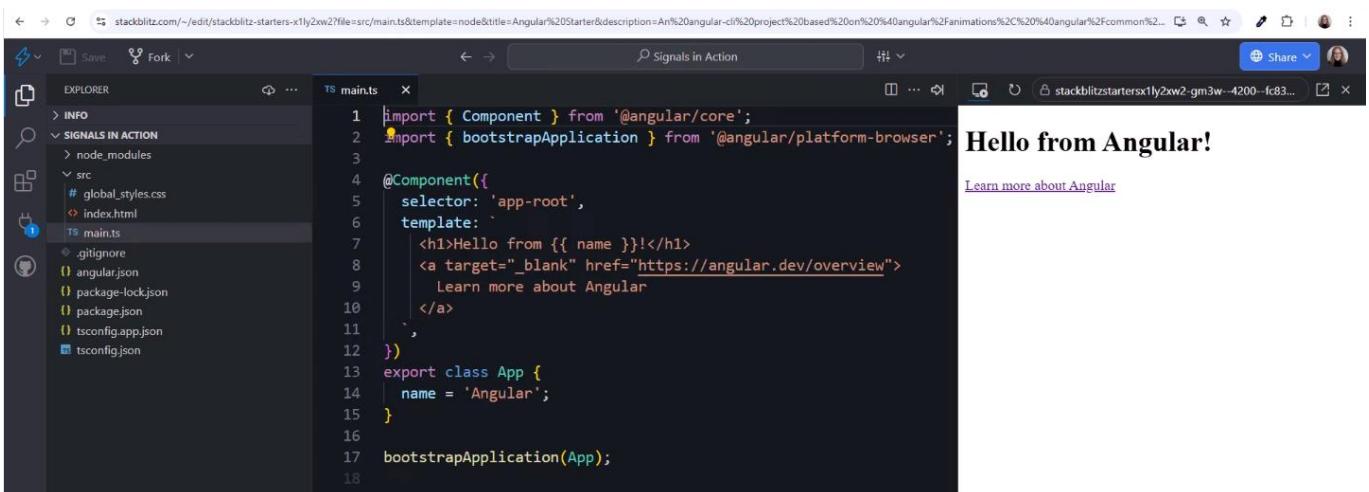
X-34 landspeeder
Quantity: 5
Vehicle: X-34 landspeeder
Price: 10550
Total: 52750



```
TS main.ts
1 import { provideHttpClient } from '@angular/common/http';
2 import { Component } from '@angular/core';
3 import { bootstrapApplication } from '@angular/platform-browser';
4 import { VehicleSelectionComponent } from './app/vehicle-selection/vehicle-selection.component';
5
6 @Component({
7   selector: 'app-root',
8   imports: [ VehicleSelectionComponent ],
9   template: `
10     <app-vehicle-selection>
11   `,
12 })
13 export class App {
14   name = 'Angular';
15 }
16
17 bootstrapApplication(App, {
18   providers: [ provideHttpClient() ]
19 });
20
```

Star Wars Vehicle Sales

X-34 landspeeder
Quantity: 5
Vehicle: X-34 landspeeder
Price: 10550
Total: 52750



```
TS main.ts
1 import { Component } from '@angular/core';
2 import { bootstrapApplication } from '@angular/platform-browser';
3
4 @Component({
5   selector: 'app-root',
6   template: `
7     <h1>Hello from {{ name }}!</h1>
8     <a target="_blank" href="https://angular.dev/overview">
9       Learn more about Angular
10     </a>
11   `,
12 })
13 export class App {
14   name = 'Angular';
15 }
16
17 bootstrapApplication(App);
```

Hello from Angular!

[Learn more about Angular](#)

The screenshot shows the StackBlitz IDE interface with the main.ts file open. The code is a basic Angular application:

```
1 import { provideHttpClient } from '@angular/common/http';
2 import { Component } from '@angular/core';
3 import { bootstrapApplication } from '@angular/platform-browser';
4
5 @Component({
6   selector: 'app-root',
7   template: `
8     <h1>Hello from {{ name }}!</h1>
9     <a href="https://angular.dev/overview">
10       Learn more about Angular
11     </a>
12   `,
13 })
14 export class App {
15   | name = 'Angular';
16 }
17
18 bootstrapApplication(App, {
19   | providers: [ provideHttpClient() ]
20 });
21
```

The right panel displays the output: "Hello from Angular!" with a link to learn more about Angular.

The screenshot shows the StackBlitz IDE interface with the vehicle.service.ts file open. The code defines a VehicleService:

```
1 import { Injectable } from '@angular/core';
2
3 @Injectable({
4   providedIn: 'root'
5 })
6 export class VehicleService {
7   | constructor() { }
8 }
```

The right panel displays the output: "Hello from Angular!" with a link to learn more about Angular.

The screenshot shows the StackBlitz IDE interface with the vehicle.service.ts file open. The code defines a VehicleService with HttpClient injection and interface definitions:

```
1 import { HttpClient } from '@angular/common/http';
2 import { Injectable, inject } from '@angular/core';
3
4 @Injectable({
5   providedIn: 'root'
6 })
7 export class VehicleService {
8   private vehicleUrl = 'https://swapi.py4e.com/api/vehicles';
9
10   private http = inject(HttpClient);
11 }
12
13 export interface VehicleResponse {
14   count: number;
15   next: string;
16   previous: string;
17   results: Vehicle[]
18 }
19 export interface Vehicle {
20   name: string;
21   cost_in_credits: number;
22 }
```

The right panel displays the output: "Hello from Angular!" with a link to learn more about Angular.

The screenshot shows a StackBlitz editor interface with the following details:

- Title Bar:** stackblitz.com/~edit/stackblitz-starters-x1ly2xw2?file=src/app/vehicle.service.ts&template=node&title=Angular%20Starter&description=An%20Angular-CLI%20based%20on%20angular%2Fanimations%2C%20angular%...
- File List:** main.ts, vehicle.service.ts
- Vehicle Service Code (vehicle.service.ts):**

```
1 import { HttpClient } from '@angular/common/http';
2 import { Injectable, computed, inject, signal } from '@angular/core';
3
4 @Injectable({
5   providedIn: 'root'
6 })
7 export class VehicleService {
8   private vehicleUrl = 'https://swapi.py4e.com/api/vehicles';
9
10  private http = inject(HttpClient);
11
12  vehicles = signal<Vehicle[]>(); // Object is possibly 'undefined'. ts(2532)
13  selectedVehicle = signal<Vehicle | undefined>();
14  quantity = signal(1); // (property) VehicleService.selectedVehicle: WritableSignal<Vehicle | undefined>
15
16  total = computed(() => this.selectedVehicle()?.cost_in_credits * this.quantity());
17  color = '';
18 }
19
20 export interface VehicleResponse {
21   count: number;
22   next: string;
23   previous: string;
24   results: Vehicle[];
25 }
26 export interface Vehicle {
27   name: string;
28   cost_in_credits: number;
29 }
```

- Right Panel:** Hello from Angular! (with a "Learn more about Angular" link)
- Bottom Status Bar:** stackblitzstartersx1ly2xw2.gm3w--4200-fc83...

stackblitz.com/~edit/stackblitz-starters-x1ly2xw2?file=src/app/vehicle.service.ts&template=node&title=Angular%20Starter&description=An%20Angular-cl%20project%20based%20on%20angular%2Fanimations%2C%20%40angular%... Share

```
TS main.ts TS vehicle.service.ts x
1 import { HttpClient } from '@angular/common/http';
2 import { Injectable, computed, inject, signal } from '@angular/core';
3
4 @Injectable({
5   providedIn: 'root'
6 })
7 export class VehicleService {
8   private vehicleUrl = 'https://swapi.py4e.com/api/vehicles';
9
10  private http = inject(HttpClient);
11
12  vehicles = signal<Vehicle[]>([]);
13  selectedVehicle = signal<Vehicle | undefined>(undefined);
14  quantity = signal(1);
15
16  total = computed(() => (this.selectedVehicle()?.cost_in_credits ?? 0) * this.quantity());
17  color = ...;
18 }
19
20 export interface VehicleResponse {
21   count: number;
22   next: string;
23   previous: string;
24   results: Vehicle[]
25 }
26 export interface Vehicle {
27   name: string;
28   cost_in_credits: number;
29 }
```

Signals in Action

Hello from Angular!

Learn more about Angular

stackblitz.com/~edit/stackblitz-starters-x1ly2xw2?file=src/app/vehicle.service.ts&template=node&title=Angular%20Starter&description=An%20Angular-cl%20project%20based%20on%20angular%2Fanimations%2C%20%40angular%... Share

```
TS main.ts TS vehicle.service.ts x
1 import { HttpClient } from '@angular/common/http';
2 import { Injectable, computed, inject, signal } from '@angular/core';
3
4 @Injectable({
5   providedIn: 'root'
6 })
7 export class VehicleService {
8   private vehicleUrl = 'https://swapi.py4e.com/api/vehicles';
9
10  private http = inject(HttpClient);
11
12  vehicles = signal<Vehicle[]>([]);
13  selectedVehicle = signal<Vehicle | undefined>(undefined);
14  quantity = signal(1);
15
16  total = computed(() => (this.selectedVehicle()?.cost_in_credits ?? 0) * this.quantity());
17  color = computed(() => this.total() > 50000 ? 'green' : 'blue');
18 }
19
20 export interface VehicleResponse {
21   count: number;
22   next: string;
23   previous: string;
24   results: Vehicle[]
25 }
26 export interface Vehicle {
27   name: string;
28   cost_in_credits: number;
29 }
```

Signals in Action

Hello from Angular!

Learn more about Angular

The screenshot shows the StackBlitz IDE interface. The left sidebar displays the project structure:

- INFO
- SIGNALS IN ACTION
- src
 - app
 - vehicle-selection
 - vehicle-selection.component.ts
 - global.styles.css
 - index.html
 - main.ts
 - .gitignore
 - angular.json
 - package-lock.json
 - package.json
 - tsconfig.app.json
 - tsconfig.json
- OUTLINE
- TIMELINE

The main editor window shows the code for `vehicle-selection.component.ts`:import { Component } from '@angular/core';
@Component({
 selector: 'app-vehicle-selection',
 imports: [],
 template:
 `

vehicle-selection works!

',
 styles: []
})
export class VehicleSelectionComponent {
}
The right panel displays the output: "Hello from Angular!" and a link to "Learn more about Angular". Below the editor is a terminal window showing build logs:~/projects/stackblitz-starters-x1ly2xw2 ls
ng g s vehicle --skip-tests
CREATE src/app/vehicle.service.ts (136 bytes)
~/projects/stackblitz-starters-x1ly2xw2 ls
ng c vehicle-selection --inline-template --inline-style --skip-tests
CREATE src/app/vehicle-selection/vehicle-selection.component.ts (236 bytes)
~/projects/stackblitz-starters-x1ly2xw2 ls
[]

The screenshot shows the StackBlitz IDE interface. The left sidebar displays the project structure:

- INFO
- SIGNALS IN ACTION
- src
 - main.ts
 - vehicle.service.ts
 - vehicle-selection.component.ts
- OUTLINE
- TIMELINE

The main editor window shows the code for `vehicle-selection.component.ts`:import { Component, inject } from '@angular/core';
import { VehicleService } from '../vehicle.service';

@Component({
 selector: 'app-vehicle-selection',
 imports: [],
 template:
 `

vehicle-selection works!

',
 styles: []
})
export class VehicleSelectionComponent {
 private vehicleService = inject(VehicleService);

 vehicles = this.vehicleService.vehicles;
 selectedVehicle = this.vehicleService.selectedVehicle;
 quantity = this.vehicleService.quantity;
 total = this.vehicleService.total;
 color = this.vehicleService.color;
}The right panel displays the output: "Hello from Angular!" and a link to "Learn more about Angular".

The screenshot shows the StackBlitz editor interface with the file `src/app/vehicle-selection/vehicle-selection.component.ts` open. The component defines a dropdown menu for selecting a vehicle from a list of Star Wars vehicles, along with input fields for quantity and total price.

```
1 import { Component, inject } from '@angular/core';
2 import { VehicleService } from './vehicle.service';
3 import { FormsModule } from '@angular/forms';
4
5 @Component({
6   selector: 'app-vehicle-selection',
7   imports: [FormsModule],
8   template: `
9     <h1>Star Wars Vehicle Sales</h1>
10    <select class="select" [(ngModel)]="selectedVehicle">
11      <option value="undefined" disabled selected>--Select a vehicle--</option>
12      @for(vehicle of vehicles()); track vehicle {
13        <option [ngValue]='vehicle'>{{ vehicle.name }}</option>
14      }
15    </select>
16    <div>Quantity: <input type='number' [(ngModel)]='quantity' ></div>
17    <div>Vehicle: {{ selectedVehicle()?.name }}</div>
18    <div>Price: {{ selectedVehicle()?.cost_in_credits }}</div>
19    <div [style.color]:'color'>Total: {{ total() }}</div>
20  `,
21  styles: ''
22 })
23 export class VehicleSelectionComponent {
24   private vehicleService = inject(VehicleService);
25
26   vehicles = this.vehicleService.vehicles;
27   selectedVehicle = this.vehicleService.selectedVehicle;
28   quantity = this.vehicleService.quantity;
29   total = this.vehicleService.total;
30 }
```

Next, we need to display this component in our application

The screenshot shows the StackBlitz editor with the file `src/main.ts` open. It includes the `VehicleSelectionComponent` and sets it as the root component for the application.

```
1 import { provideHttpClient } from '@angular/common/http';
2 import { Component } from '@angular/core';
3 import { bootstrapApplication } from '@angular/platform-browser';
4
5 @Component({
6   selector: 'app-root',
7   template: `
8     <h1>Hello from {{ name }}!</h1>
9     <a target="_blank" href="https://angular.dev/overview">
10       Learn more about Angular
11     </a>
12   `,
13 })
14 export class App {
15   name = 'Angular';
16 }
17
18 bootstrapApplication(App, {
19   providers: [provideHttpClient()]
20 });
```

The screenshot shows the StackBlitz editor with the file `src/main.ts` open. A yellow arrow points to a red error box containing a warning message about the `<app-vehicle-selection/>` element.

NG8001: 'app-vehicle-selection' is not a known element:

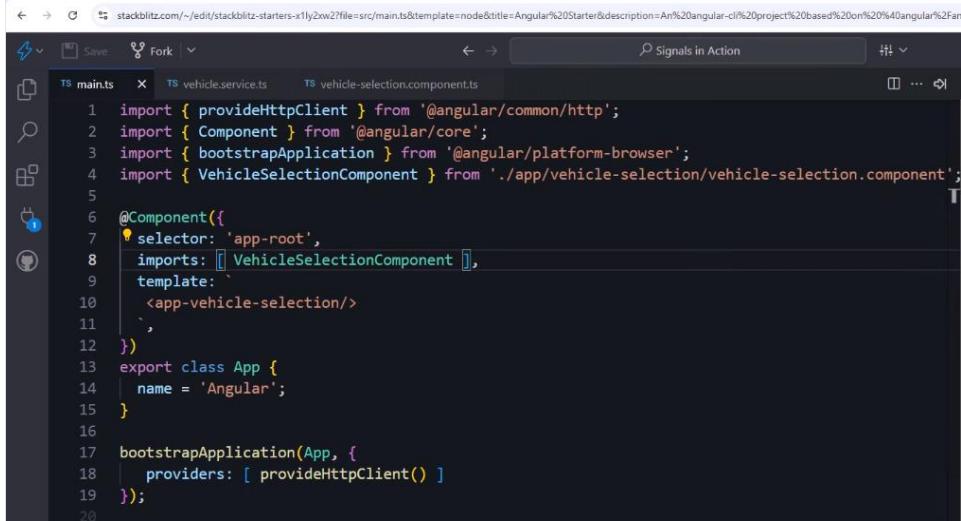
- If 'app-vehicle-selection' is an Angular component, then verify that it is included in the '`@Component.imports`' of this component.
- If 'app-vehicle-selection' is a Web Component then add '`CUSTOM_ELEMENTS_SCHEMA`' to the '`@Component.schemas`' of this component to suppress this message.

src/main.ts:8:3

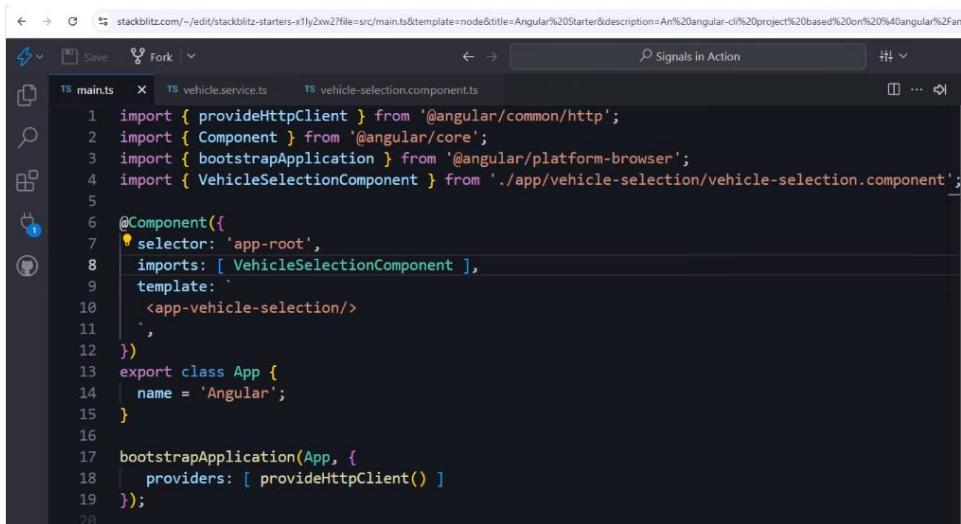
Click outside, press `Esc` key, or fix the code to dismiss.

```
1 import { provideHttpClient } from '@angular/common/http';
2 import { Component } from '@angular/core';
3 import { bootstrapApplication } from '@angular/platform-browser';
4
5 @Component({
6   selector: 'app-root',
7   template: `
8     <app-vehicle-selection/>
9   `,
10 })
11 export class App {
12   name = 'Angular';
13 }
14
15 bootstrapApplication(App, {
16   providers: [provideHttpClient()]
17 });
```

This is because the default in Angular v19 is standalone components, each component must import everything it needs

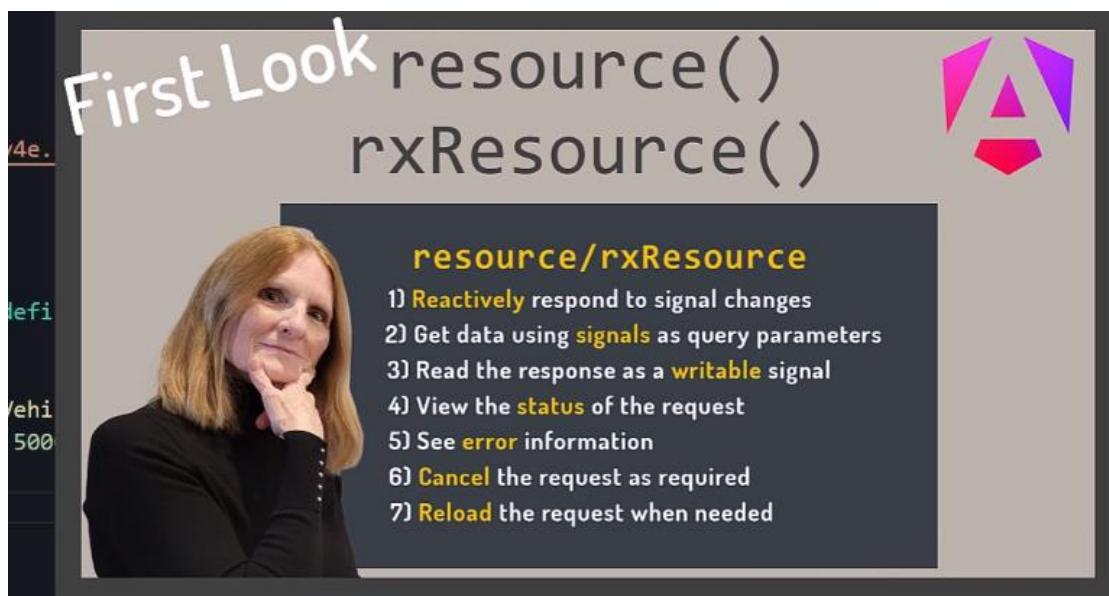


StackBlitz screenshot showing the initial state of the application. The code in main.ts imports VehicleSelectionComponent from vehicle-selection.component.ts. The component template contains a single <app-vehicle-selection> tag.



StackBlitz screenshot showing the state after removing the app- prefix from the component selector. The dropdown menu now shows 'Select a vehicle' instead of 'Vehicle'.

We now need to retrieve some data using the HTTP service as below using the **rxResource API**



stackblitz.com/~edit/stackblitz-starters-x1ly2xw2?file=src/app/vehicle.service.ts&template=node&title=Angular%20Starter&description=An%20angular-cl%20project%20based%20on%20angular%2fanimations%2C%20%40angular%...

Save Fork Signals in Action Share

TS main.ts TS vehicle.service.ts TS vehicle-selection.component.ts

```
1 import { HttpClient } from '@angular/common/http';
2 import { Injectable, computed, inject, signal } from '@angular/core';
3
4 @Injectable({
5   providedIn: 'root'
6 })
7 export class VehicleService {
8   private vehicleUrl = 'https://swapi.py4e.com/api/vehicles';
9
10  private http = inject(HttpClient);
11
12  vehicles = signal<Vehicle[]>([]);
13  selectedVehicle = signal<Vehicle | undefined>(undefined);
14  quantity = signal(1);
15
16  total = computed(() => (this.selectedVehicle()?.cost_in_credits ?? 0) * this.quantity());
17  color = computed(() => this.total() > 50000 ? 'green' : 'blue');
18 }
19
20 export interface VehicleResponse {
21   count: number;
22   next: string;
23   previous: string;
24   results: Vehicle[]
25 }
26 export interface Vehicle {
27   name: string;
28   cost_in_credits: number;
29 }
```

Star Wars Vehicle Sales

-Select a vehicle--
Quantity: 1
Vehicle:
Price:
Total: 0

stackblitz.com/~edit/stackblitz-starters-x1ly2xw2?file=src/app/vehicle.service.ts&template=node&title=Angular%20Starter&description=An%20angular-cl%20project%20based%20on%20angular%2fanimations%2C%20%40angular%...

Save Fork Signals in Action Share

TS main.ts TS vehicle.service.ts TS vehicle-selection.component.ts

```
1 import { HttpClient } from '@angular/common/http';
2 import { Injectable, computed, inject, signal } from '@angular/core';
3 import { rxResource } from '@angular/core/rxjs-interop';
4
5 @Injectable({
6   providedIn: 'root'
7 })
8 export class VehicleService {
9   private vehicleUrl = 'https://swapi.py4e.com/api/vehicles';
10
11  private http = inject(HttpClient);
12
13  vehicles = signal<Vehicle[]>([]);
14  selectedVehicle = signal<Vehicle | undefined>(undefined);
15  quantity = signal(1);
16
17  total = computed(() => (this.selectedVehicle()?.cost_in_credits ?? 0) * this.quantity());
18  color = computed(() => this.total() > 50000 ? 'green' : 'blue');
19
20  vehicleResource = rxResource({
21    loader: () => this.http.get([this.vehicleUrl])
22  });
23 }
24
25 export interface VehicleResponse {
26   count: number;
27   next: string;
28   previous: string;
29   results: Vehicle[]
30 }
```

TS2554: Expected 1-2 arguments, but got 0.
src/app/vehicle.service.ts:21:30

Click outside, press Esc key, or fix the code to dismiss.

stackblitz.com/~edit/stackblitz-starters-x1ly2xw2?file=src/app/vehicle.service.ts&template=node&title=Angular%20Starter&description=An%20angular-di%20project%20based%20on%20angular%2Fanimations%2C%20%40angular%...

Signals in Action

Share

TS main.ts TS vehicle.service.ts TS vehicle-selection.component.ts

```
8 export class VehicleService {
9   private vehicleUrl = 'https://swapi.py4e.com/api/vehicles';
10
11   private http = inject(HttpClient);
12
13   vehicles = signal<Vehicle[]>([]);
14   selectedVehicle = signal<Vehicle | undefined>(undefined);
15   quantity = signal(1);
16
17   total = computed(() => (this.selectedVehicle()?.cost_in_credits ?? 0) * this.quantity());
18   color = computed(() => this.total() > 50000 ? 'green' : 'blue');
19
20   vehicleResource = rxResource({
21     loader: () => this.http.get<VehicleResponse>(this.vehicleUrl)
22   });
23 }
24
25 export interface VehicleResponse {
26   count: number;
27   next: string;
28   previous: string;
29   results: Vehicle[]
30 }
31 export interface Vehicle {
32   name: string;
33   cost_in_credits: number;
34 }
```

Star Wars Vehicle Sales

-Select a vehicle--

Quantity: 1

Vehicle:

Price:

Total: 0

stackblitz.com/~edit/stackblitz-starters-x1ly2xw2?file=src/app/vehicle.service.ts&template=node&title=Angular%20Starter&description=An%20angular-di%20project%20based%20on%20angular%2Fanimations%2C%20%40angular%...

Signals in Action

Share

TS main.ts TS vehicle.service.ts TS vehicle-selection.component.ts

```
8 })
9 export class VehicleService {
10   private vehicleUrl = 'https://swapi.py4e.com/api/vehicles';
11
12   private http = inject(HttpClient);
13
14   vehicles = signal<Vehicle[]>([]);
15   selectedVehicle = signal<Vehicle | undefined>(undefined);
16   quantity = signal(1);
17
18   total = computed(() => (this.selectedVehicle()?.cost_in_credits ?? 0) * this.quantity());
19   color = computed(() => this.total() > 50000 ? 'green' : 'blue');
20
21   vehicleResource = rxResource({
22     loader: () => this.http.get<VehicleResponse>(this.vehicleUrl).pipe(
23       map((vr) => vr.results)
24     )
25   );
26 }
27
28 export interface VehicleResponse {
29   count: number;
30   next: string;
31   previous: string;
32   results: Vehicle[]
33 }
34 export interface Vehicle {
35   name: string;
36   cost_in_credits: number;
```

Star Wars Vehicle Sales

-Select a vehicle--

Quantity: 1

Vehicle:

Price:

Total: 0

stackblitz.com/~edit/stackblitz-starters-x1ly2xw2?file=src/app/vehicle.service.ts&template=node&title=Angular%20Starter&description=An%20angular-di%20project%20based%20on%20angular%2Fanimations%2C%20%40angular%...

Save Fork Signals in Action Share

```
8 })
9 export class VehicleService {
10   private vehicleUrl = 'https://swapi.py4e.com/api/vehicles';
11
12   private http = inject(HttpClient);
13
14   selectedVehicle = signal<Vehicle | undefined>(undefined);
15   quantity = signal(1);
16
17   total = computed(() => (this.selectedVehicle()?.cost_in_credits ?? 0) * this.quantity());
18   color = computed(() => this.total() > 50000 ? 'green' : 'blue');
19
20   vehicleResource = rxResource({
21     loader: () => this.http.get<VehicleResponse>(this.vehicleUrl).pipe(
22       map(vr => vr.results)
23     )
24   });
25   vehicles = computed(() => this.vehicleResource.value());
26 }
27
28 export interface VehicleResponse {
29   count: number;
30   next: string;
31   previous: string;
32   results: Vehicle[]
33 }
34 export interface Vehicle {
35   name: string;
36   cost_in_credits: number;
```

Star Wars Vehicle Sales

-Select a vehicle--
Quantity: 1
Vehicle:
Price:
Total: 0

stackblitz.com/~edit/stackblitz-starters-x1ly2xw2?file=src/app/vehicle.service.ts&template=node&title=Angular%20Starter&description=An%20angular-di%20project%20based%20on%20angular%2Fanimations%2C%20%40angular%...

Save Fork Signals in Action Share

```
8 })
9 export class VehicleService {
10   private vehicleUrl = 'https://swapi.py4e.com/api/vehicles';
11
12   private http = inject(HttpClient);
13
14   selectedVehicle = signal<Vehicle | undefined>(undefined);
15   quantity = signal(1);
16
17   total = computed(() => (this.selectedVehicle()?.cost_in_credits ?? 0) * this.quantity());
18   color = computed(() => this.total() > 50000 ? 'green' : 'blue');
19
20   vehicleResource = rxResource({
21     loader: () => this.http.get<VehicleResponse>(this.vehicleUrl).pipe(
22       map(vr => vr.results)
23     )
24   });
25   vehicles = computed(() => this.vehicleResource.value()); 
26 }
27
28 export interface VehicleResponse {
29   count: number;
30   next: string;
31   previous: string;
32   results: Vehicle[]
33 }
34 export interface Vehicle {
35   name: string;
36   cost_in_credits: number;
```

Star Wars Vehicle Sales

-Select a vehicle--
Quantity: 1
Vehicle:
Price:
Total: 0

stackblitz.com/~edit/stackblitz-starters-x1ly2xw2?file=src/app/vehicle.service.ts&template=node&title=Angular%20Starter&description=An%20Angular-di%20project%20based%20on%20angular%2fanimations%2C%20angular%...

Save Fork Signals in Action

TS main.ts TS vehicle.service.ts vehicle-selection.component.ts

```
8  })
9  export class VehicleService {
10    private vehicleUrl = 'https://swapi.py4e.com/api/vehicles';
11
12    private http = inject(HttpClient);
13
14    selectedVehicle = signal<Vehicle | undefined>(undefined);
15    quantity = signal(1);
16
17    total = computed(() => (this.selectedVehicle()?.cost_in_credits ?? 0) * this.quantity());
18    color = computed(() => this.total() > 50000 ? 'green' : 'blue');
19
20    vehicleResource = rxResource({
21      loader: () => this.http.get<VehicleResponse>(this.vehicleUrl).pipe(
22        map(vr => vr.results)
23      )
24    });
25    vehicles = computed(() => this.vehicleResource.value() ?? [] as Vehicle[]);
26  }
27
28  export interface VehicleResponse {
29    count: number;
30    next: string;
31    previous: string;
32    results: Vehicle[]
33  }
34  export interface Vehicle {
35    name: string;
36    cost_in_credits: number;
```

Star Wars Vehicle Sales

--Select a vehicle--
Quantity: 1
Vehicle:
Price:
Total: 0

stackblitz.com/~edit/stackblitz-starters-x1ly2xw2?file=src/app/vehicle.service.ts&template=node&title=Angular%20Starter&description=An%20Angular-di%20project%20based%20on%20angular%2fanimations%2C%20angular%...

Save Fork Signals in Action

TS main.ts TS vehicle.service.ts vehicle-selection.component.ts

```
8  })
9  export class VehicleService {
10    private vehicleUrl = 'https://swapi.py4e.com/api/vehicles';
11
12    private http = inject(HttpClient);
13
14    selectedVehicle = signal<Vehicle | undefined>(undefined);
15    quantity = signal(1);
16
17    total = computed(() => (this.selectedVehicle()?.cost_in_credits ?? 0) * this.quantity());
18    color = computed(() => this.total() > 50000 ? 'green' : 'blue');
19
20    vehicleResource = rxResource({
21      loader: () => this.http.get<VehicleResponse>(this.vehicleUrl).pipe(
22        map(vr => vr.results)
23      )
24    });
25    vehicles = computed(() => this.vehicleResource.value() ?? [] as Vehicle[]);
26  }
27
28  export interface VehicleResponse {
29    count: number;
30    next: string;
31    previous: string;
32    results: Vehicle[]
33  }
34  export interface Vehicle {
35    name: string;
36    cost_in_credits: number;
```

Star Wars Vehicle Sales

--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

stackblitz.com/~edit/stackblitz-starters-x1ly2xw2?file=src/app/vehicle.service.ts&template=node&title=Angular%20Starter&description=An%20Angular-dI%20project%20based%20on%20angular%2fanimations%2C%20angular%...

Signals in Action

Share

TS main.ts TS vehicle.service.ts TS vehicle-selection.component.ts

```
8 })
9 export class VehicleService {
10   private vehicleUrl = 'https://swapi.py4e.com/api/vehicles';
11
12   private http = inject(HttpClient);
13
14   selectedVehicle = signal<Vehicle | undefined>(undefined);
15   quantity = signal(1);
16
17   total = computed(() => (this.selectedVehicle()?.cost_in_credits ?? 0) * this.quantity());
18   color = computed(() => this.total() > 50000 ? 'green' : 'blue');
19
20   vehicleResource = rxResource({
21     loader: () => this.http.get<VehicleResponse>(this.vehicleUrl).pipe(
22       map(vr => vr.results)
23     )
24   });
25   vehicles = computed(() => this.vehicleResource.value() ?? [] as Vehicle[]);
26 }
27
28 export interface VehicleResponse {
29   count: number;
30   next: string;
31   previous: string;
32   results: Vehicle[]
33 }
34 export interface Vehicle {
35   name: string;
36   cost_in_credits: number;
```

Star Wars Vehicle Sales

X-34 landspeeder

Quantity: 1

Vehicle: X-34 landspeeder

Price: 10550

Total: 10550

stackblitz.com/~edit/stackblitz-starters-x1ly2xw2?file=src/app/vehicle.service.ts&template=node&title=Angular%20Starter&description=An%20Angular-dI%20project%20based%20on%20angular%2fanimations%2C%20angular%...

Signals in Action

Share

TS main.ts TS vehicle.service.ts TS vehicle-selection.component.ts

```
8 })
9 export class VehicleService {
10   private vehicleUrl = 'https://swapi.py4e.com/api/vehicles';
11
12   private http = inject(HttpClient);
13
14   selectedVehicle = signal<Vehicle | undefined>(undefined);
15   quantity = signal(1);
16
17   total = computed(() => (this.selectedVehicle()?.cost_in_credits ?? 0) * this.quantity());
18   color = computed(() => this.total() > 50000 ? 'green' : 'blue');
19
20   vehicleResource = rxResource({
21     loader: () => this.http.get<VehicleResponse>(this.vehicleUrl).pipe(
22       map(vr => vr.results)
23     )
24   });
25   vehicles = computed(() => this.vehicleResource.value() ?? [] as Vehicle[]);
26 }
27
28 export interface VehicleResponse {
29   count: number;
30   next: string;
31   previous: string;
32   results: Vehicle[]
33 }
34 export interface Vehicle {
35   name: string;
36   cost_in_credits: number;
```

Star Wars Vehicle Sales

X-34 landspeeder

Quantity: 3

Vehicle: X-34 landspeeder

Price: 10550

Total: 31650

Stackblitz Editor

```
8  })
9  export class VehicleService {
10    private vehicleUrl = 'https://swapi.py4e.com/api/vehicles';
11
12    private http = inject(HttpClient);
13
14    selectedVehicle = signal<Vehicle | undefined>(undefined);
15    quantity = signal(1);
16
17    total = computed(() => (this.selectedVehicle()?.cost_in_credits ?? 0) * this.quantity());
18    color = computed(() => this.total() > 50000 ? 'green' : 'blue');
19
20    vehicleResource = rxResource({
21      loader: () => this.http.get<VehicleResponse>(this.vehicleUrl).pipe(
22        map(vr => vr.results)
23      )
24    );
25    vehicles = computed(() => this.vehicleResource.value() ?? [] as Vehicle[]);
26  }
27
28  export interface VehicleResponse {
29    count: number;
30    next: string;
31    previous: string;
32    results: Vehicle[]
33  }
34  export interface Vehicle {
35    name: string;
36    cost_in_credits: number;
}
```

Star Wars Vehicle Sales

X-34 landspeeder
Quantity: 5
Vehicle: X-34 landspeeder
Price: 10550
Total: 52750

Stackblitz Editor

```
8  })
9  export class VehicleService {
10    private vehicleUrl = 'https://swapi.py4e.com/api/vehicles';
11
12    private http = inject(HttpClient);
13
14    selectedVehicle = signal<Vehicle | undefined>(undefined);
15    quantity = signal(1);
16
17    total = computed(() => (this.selectedVehicle()?.cost_in_credits ?? 0) * this.quantity());
18    color = computed(() => this.total() > 50000 ? 'green' : 'blue');
19
20    vehicleResource = rxResource({
21      loader: () => this.http.get<VehicleResponse>(this.vehicleUrl).pipe(
22        map(vr => vr.results)
23      )
24    );
25    vehicles = computed(() => this.vehicleResource.value() ?? [] as Vehicle[]);
26  }
27
28  export interface VehicleResponse {
29    count: number;
30    next: string;
31    previous: string;
32    results: Vehicle[]
33  }
34  export interface Vehicle {
35    name: string;
36    cost_in_credits: number;
}
```

Star Wars Vehicle Sales

X-34 landspeeder
--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

```

8  })
9  export class VehicleService {
10    private vehicleUrl = 'https://swapi.py4e.com/api/vehicles';
11
12    private http = inject(HttpClient);
13
14    selectedVehicle = signal<Vehicle | undefined>(undefined);
15    quantity = signal(1);
16
17    total = computed(() => (this.selectedVehicle()?.cost_in_credits ?? 0) * this.quantity());
18    color = computed(() => this.total() > 50000 ? 'green' : 'blue');
19
20    vehicleResource = rxResource({
21      loader: () => this.http.get<VehicleResponse>(this.vehicleUrl).pipe(
22        map(vr => vr.results)
23      )
24    });
25    vehicles = computed(() => this.vehicleResource.value() ?? [] as Vehicle[]);
26  }
27
28  export interface VehicleResponse {
29    count: number;
30    next: string;
31    previous: string;
32    results: Vehicle[]
33  }
34  export interface Vehicle {
35    name: string;
36    cost_in_credits: number;

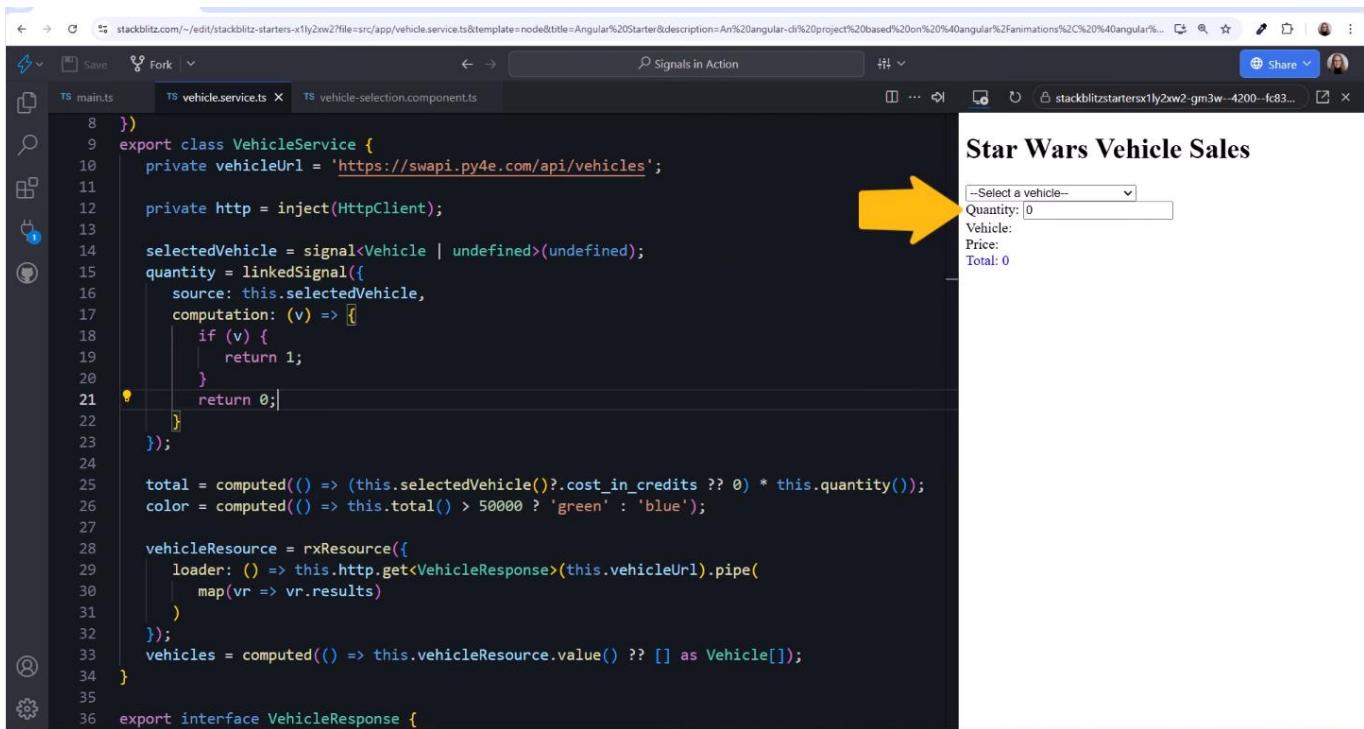
```

But we want the quantity to reset back to 1 when the user selects another vehicle. We use a **LinkedSignal** whenever we want the value of a signal to reset based on the value of another signal.

```

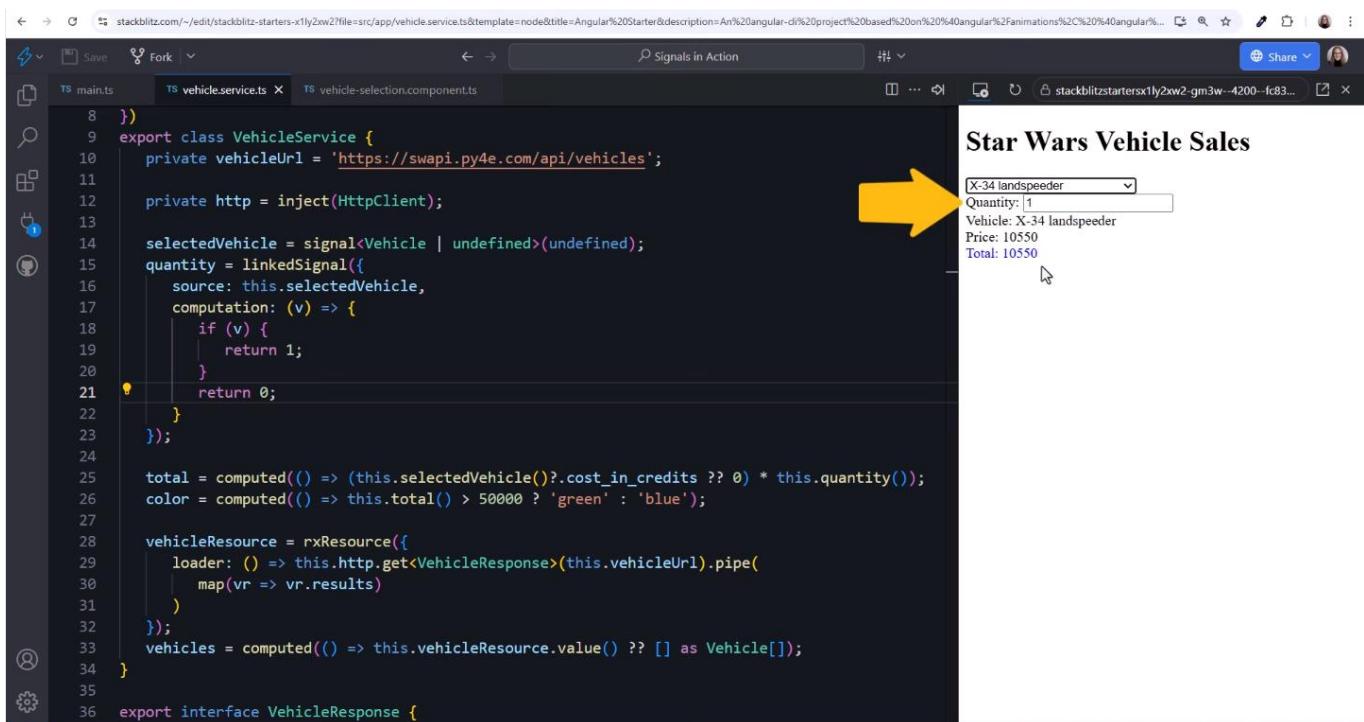
8  })
9  export class VehicleService {
10    private vehicleUrl = 'https://swapi.py4e.com/api/vehicles';
11
12    private http = inject(HttpClient);
13
14    selectedVehicle = signal<Vehicle | undefined>(undefined);
15    quantity = linkedSignal({
16      source: this.selectedVehicle,
17      computation: (v) => [
18        if (v) {
19          return 1;
20        }
21        return 0;
22      ]
23    });
24
25    total = computed(() => (this.selectedVehicle()?.cost_in_credits ?? 0) * this.quantity());
26    color = computed(() => this.total() > 50000 ? 'green' : 'blue');
27
28    vehicleResource = rxResource({
29      loader: () => this.http.get<VehicleResponse>(this.vehicleUrl).pipe(
30        map(vr => vr.results)
31      )
32    });
33    vehicles = computed(() => this.vehicleResource.value() ?? [] as Vehicle[]);
34  }
35
36  export interface VehicleResponse {

```



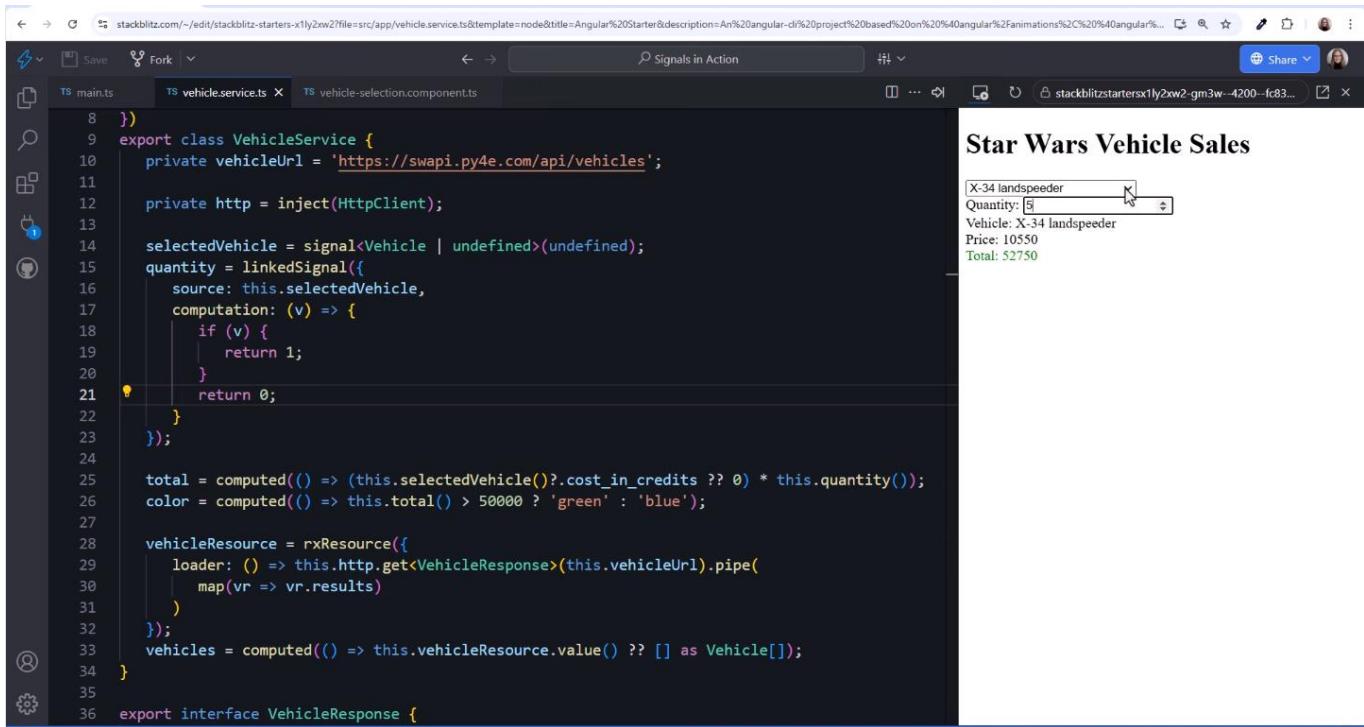
StackBlitz Editor showing the `vehicle.service.ts` file:

```
8 })
9 export class VehicleService {
10   private vehicleUrl = 'https://swapi.py4e.com/api/vehicles';
11
12   private http = inject(HttpClient);
13
14   selectedVehicle = signal<Vehicle | undefined>(undefined);
15   quantity = linkedSignal({
16     source: this.selectedVehicle,
17     computation: (v) => [
18       if (v) {
19         return 1;
20       }
21       return 0;
22     ]
23   });
24
25   total = computed(() => (this.selectedVehicle()?.cost_in_credits ?? 0) * this.quantity());
26   color = computed(() => this.total() > 50000 ? 'green' : 'blue');
27
28   vehicleResource = rxResource({
29     loader: () => this.http.get<VehicleResponse>(this.vehicleUrl).pipe(
30       map(vr => vr.results)
31     )
32   );
33   vehicles = computed(() => this.vehicleResource.value() ?? [] as Vehicle[]);
34 }
35
36 export interface VehicleResponse {
```



StackBlitz Editor showing the `vehicle.service.ts` file:

```
8 })
9 export class VehicleService {
10   private vehicleUrl = 'https://swapi.py4e.com/api/vehicles';
11
12   private http = inject(HttpClient);
13
14   selectedVehicle = signal<Vehicle | undefined>(undefined);
15   quantity = linkedSignal({
16     source: this.selectedVehicle,
17     computation: (v) => [
18       if (v) {
19         return 1;
20       }
21       return 0;
22     ]
23   });
24
25   total = computed(() => (this.selectedVehicle()?.cost_in_credits ?? 0) * this.quantity());
26   color = computed(() => this.total() > 50000 ? 'green' : 'blue');
27
28   vehicleResource = rxResource({
29     loader: () => this.http.get<VehicleResponse>(this.vehicleUrl).pipe(
30       map(vr => vr.results)
31     )
32   );
33   vehicles = computed(() => this.vehicleResource.value() ?? [] as Vehicle[]);
34 }
35
36 export interface VehicleResponse {
```



Star Wars Vehicle Sales

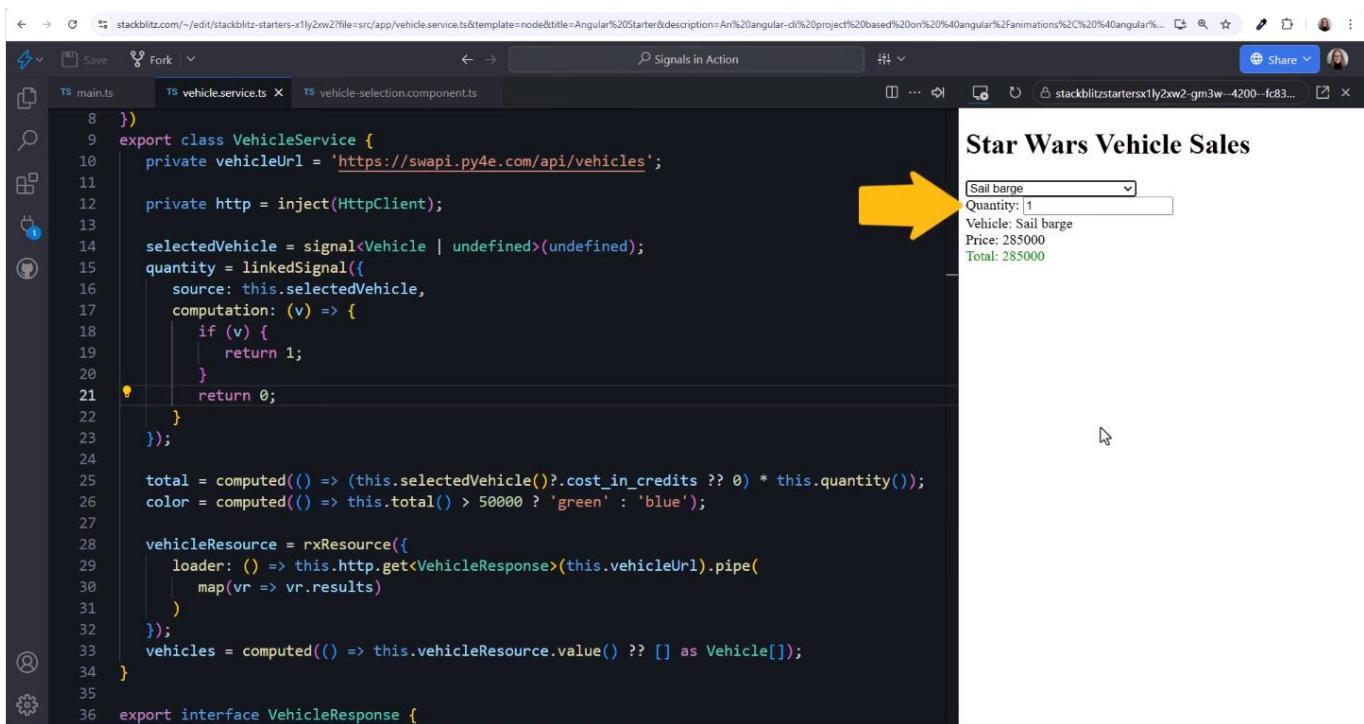
X-34 landspeeder

Quantity: 5

Vehicle: X-34 landspeeder

Price: 10550

Total: 52750



Star Wars Vehicle Sales

Sail barge

Quantity: 1

Vehicle: Sail barge

Price: 285000

Total: 285000

It works

stackblitz.com/~edit/stackblitz-starters-x1ly2xw2?file=src/app/vehicle-selection/vehicle-selection.component.ts&template=node&title=Angular%20Starter&description=An%20angular-cl%20project%20based%20on%20angular%2Fa...

Save Fork Signals in Action Share

TS main.ts TS vehicle.service.ts TS vehicle-selection.component.ts

```
1 import { Component, inject } from '@angular/core';
2 import { VehicleService } from '../vehicle.service';
3 import { FormsModule } from '@angular/forms';

4
5 @Component({
6   selector: 'app-vehicle-selection',
7   imports: [FormsModule],
8   template: `
9     <h1>Star Wars Vehicle Sales</h1>
10    <select class="select" [(ngModel)]="selectedVehicle">
11      <option value="undefined" disabled selected>--Select a vehicle--</option>
12      <for( vehicle of vehicles(); track vehicle) {
13        <option [ngValue]="'vehicle'">{{ vehicle.name }}</option>
14      }
15    </select>
16    <div>Quantity: <input type='number' [(ngModel)]='quantity' ></div>
17    <div>Vehicle: {{ selectedVehicle()?.name }}</div>
18    <div>Price: {{ selectedVehicle()?.cost_in_credits }}</div>
19    <div [style.color]="'color'">Total: {{ total() }}</div>
20  `,
21  styles: ``
22 })
23 export class VehicleSelectionComponent {
24   private vehicleService = inject(VehicleService);
25
26   vehicles = this.vehicleService.vehicles;
27   selectedVehicle = this.vehicleService.selectedVehicle;
28   quantity = this.vehicleService.quantity;
29   total = this.vehicleService.total;
30 }
```

Star Wars Vehicle Sales

Sail barge
Quantity: 1
Vehicle: Sail barge
Price: 285000
Total: 285000

stackblitz.com/~edit/stackblitz-starters-x1ly2xw2?file=src/app/vehicle-selection/vehicle-selection.component.ts&template=node&title=Angular%20Starter&description=An%20angular-cl%20project%20based%20on%20angular%2Fa...

Save Fork Signals in Action Share

TS main.ts TS vehicle.service.ts TS vehicle-selection.component.ts

```
5 @Component({
6   selector: 'app-vehicle-selection',
7   imports: [FormsModule],
8   template: `
9     <h1>Star Wars Vehicle Sales</h1>
10    <select class="select" [(ngModel)]="selectedVehicle">
11      <option value="undefined" disabled selected>--Select a vehicle--</option>
12      <for( vehicle of vehicles(); track vehicle) {
13        <option [ngValue]="'vehicle'">{{ vehicle.name }}</option>
14      }
15    </select>
16    <div>Quantity: <input type='number' [(ngModel)]='quantity' ></div>
17    <div>Vehicle: {{ selectedVehicle()?.name }}</div>
18    <div>Price: {{ selectedVehicle()?.cost_in_credits }}</div>
19    <div [style.color]="'color'">Total: {{ total() }}</div>
20  `,
21  styles: ``
```

```
22 })
23 export class VehicleSelectionComponent {
24   private vehicleService = inject(VehicleService);
25
26   vehicles = this.vehicleService.vehicles;
27   selectedVehicle = this.vehicleService.selectedVehicle;
28   quantity = this.vehicleService.quantity;
29   total = this.vehicleService.total;
30   color = this.vehicleService.color;
31 }
```

Star Wars Vehicle Sales

Sail barge
Quantity: 1
Vehicle: Sail barge
Price: 285000
Total: 285000