

# First Look at NgRx



**Duncan Hunter**

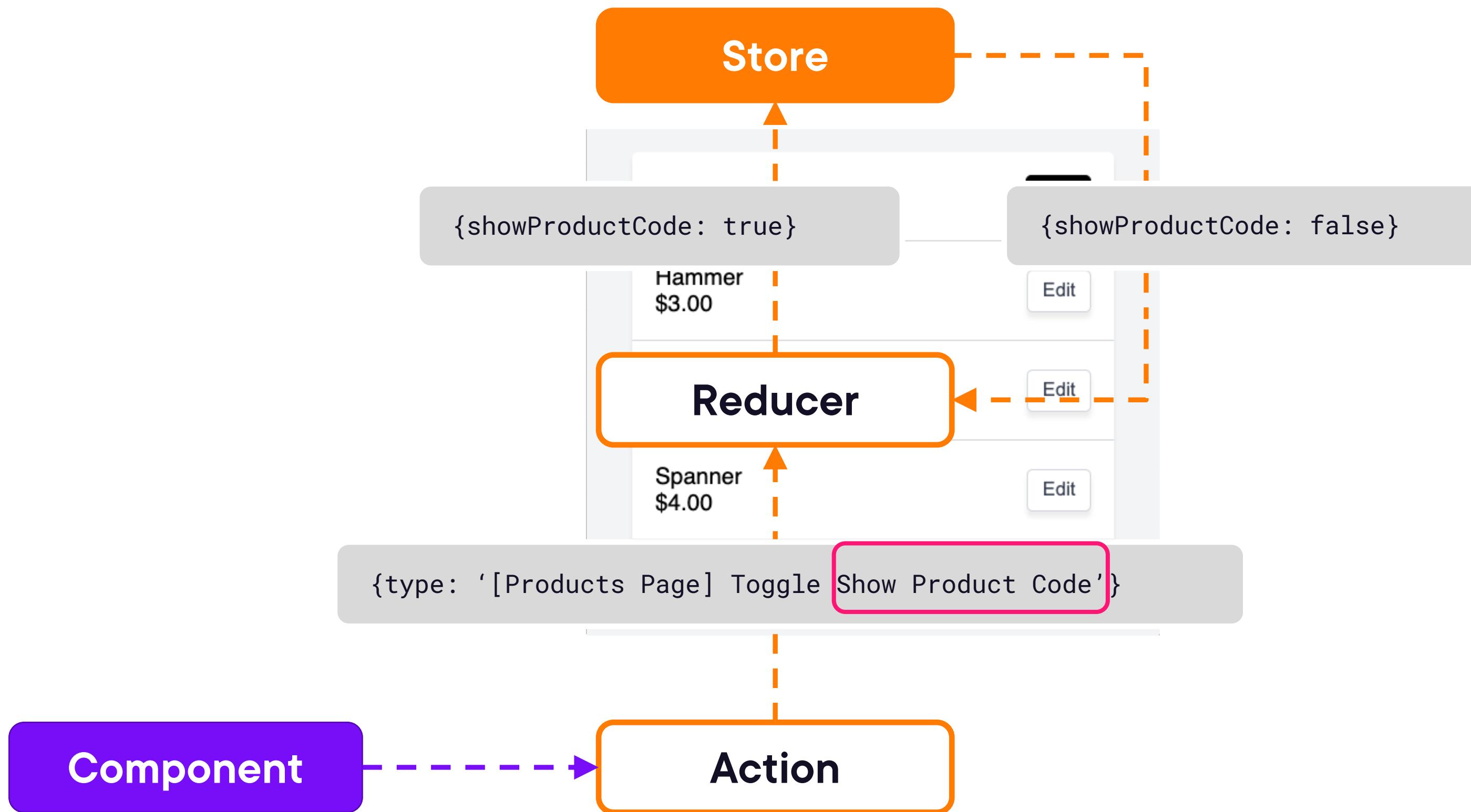
Lead Frontend Developer

@dunchunter | [duncanhunter.com.au](http://duncanhunter.com.au)

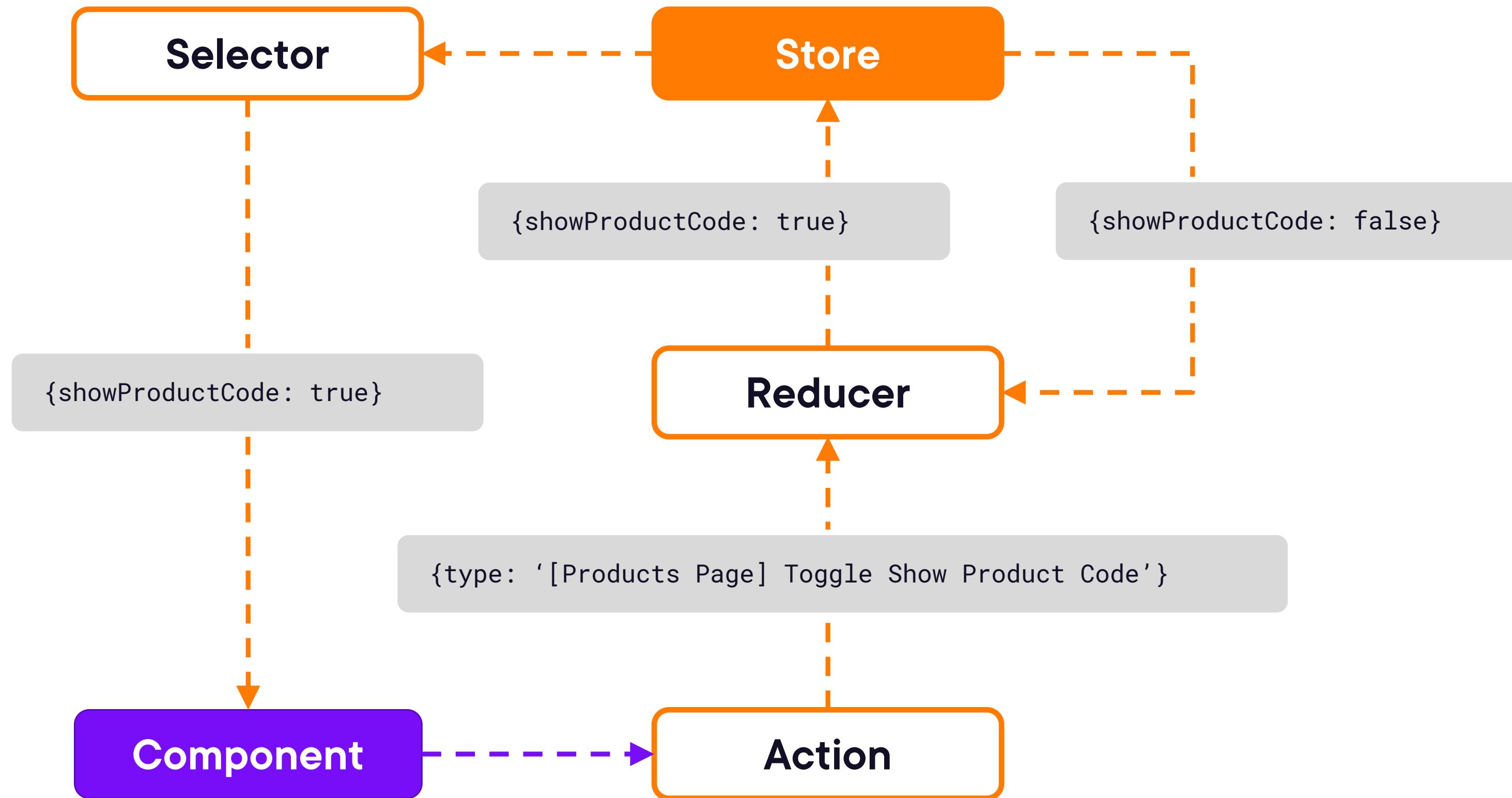




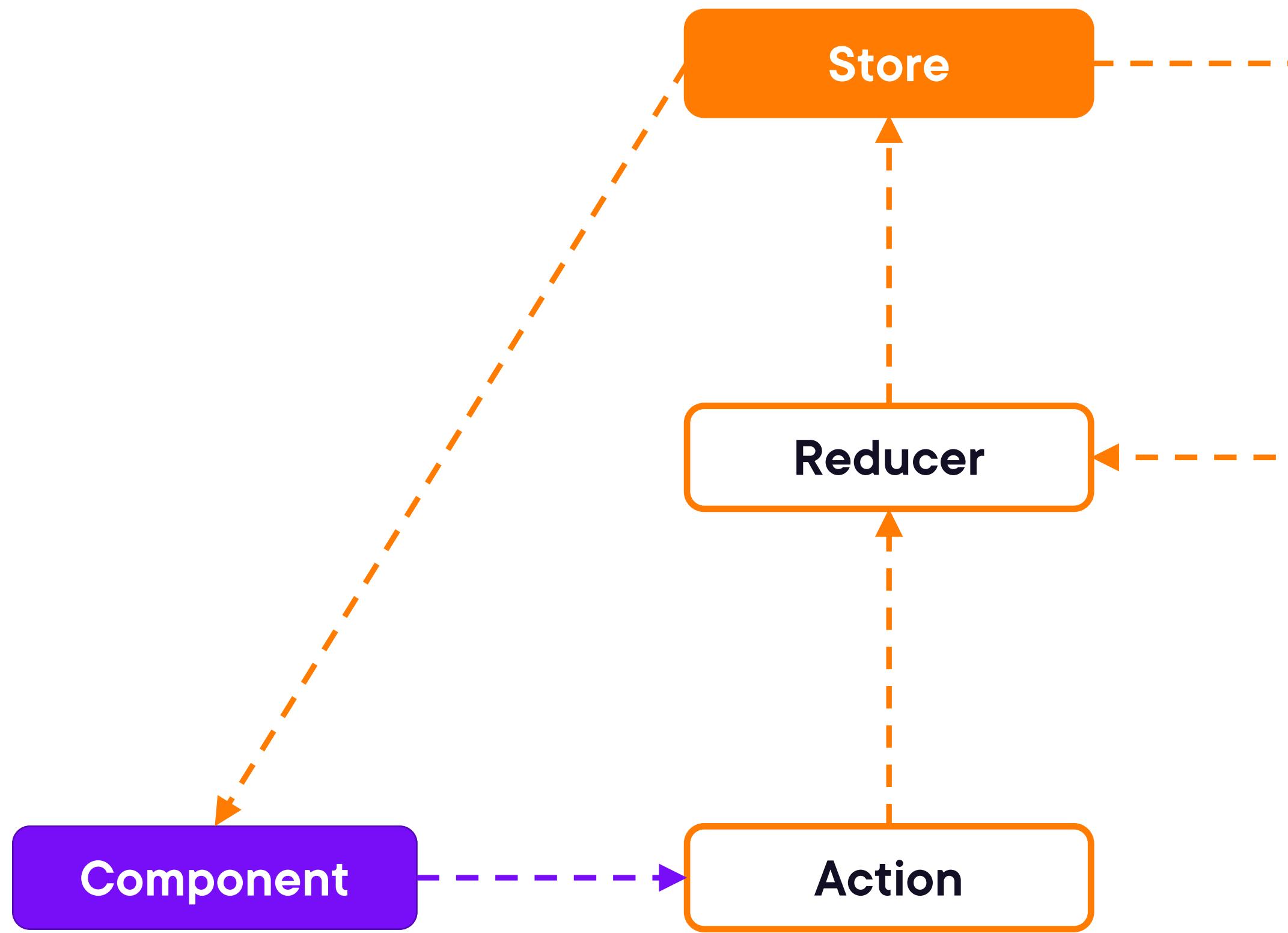
# State Lifecycle – Toggle Show Product Code



# State Lifecycle – Toggle Show Product Code



# State Lifecycle – Toggle Show Product Code



# Module Introduction



**Set up the sample application**

**Install and initialize `@ngrx/store`**

**Define the state and actions**

**Build a reducer to process actions and set the store state**

**Dispatch an action to change state**

**Subscribe to state change notifications**



# Demo



## Setting up the sample application



# State Lifecycle – Toggle Show Product Code

Store

```
{  
  products: {  
    showProductCode: true,  
    products: [{id: 1, name: 'Hammer'}]  
  },  
  auth: {  
    userId: 1,  
    email: user@me.com  
  }  
}
```



# Eager Global and Lazy Feature State

## AppModule.ts

```
import { StoreModule } from '@ngrx/store';

@NgModule({
  imports: [
    StoreModule.forRoot({})
  ]
})
export class AppModule {}
```

## ProductsModule.ts

```
import { StoreModule } from '@ngrx/store';
import { productsReducer } from 'products.reducer';

@NgModule({
  imports: [
    StoreModule.forFeature('products', productsReducer)
  ]
})
export class ProductsModule {}
```



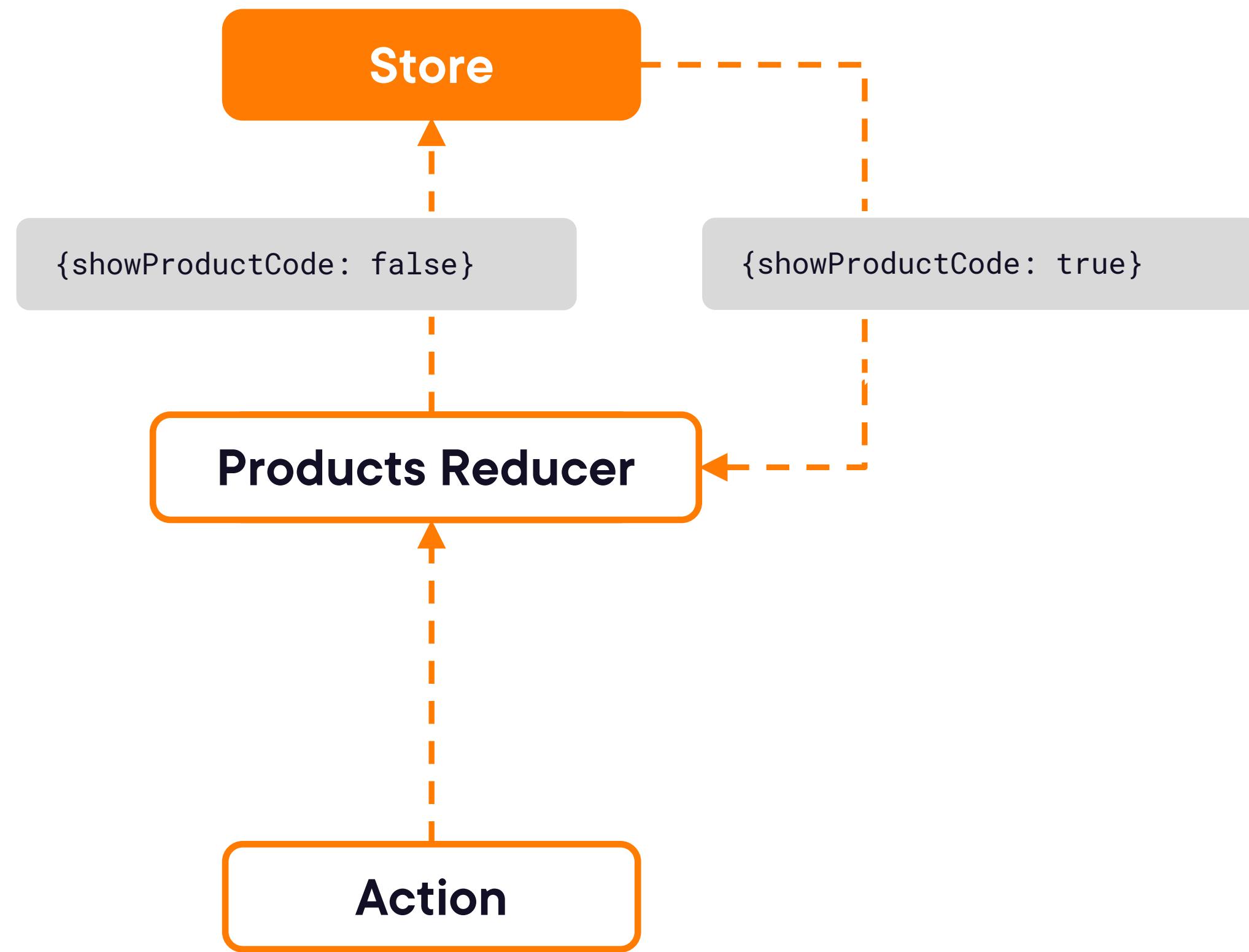
# Demo



## Install and initialize the store



# State Lifecycle – Toggle Show Product Code



# Products Reducer

```
import { createAction, createReducer, on } from '@ngrx/store';

export const productsReducer = createReducer(
  { showProductCode: true },
  on(
    )
  );

```



# Products Reducer

```
import { createAction, createReducer, on } from '@ngrx/store';

export const productsReducer = createReducer(
  { showProductCode: true },
  on(createAction('[Products Page] Toggle Show Product Code'),
    )
);
```



# Products Reducer

```
import { createAction, createReducer, on } from '@ngrx/store';

export const productsReducer = createReducer(
  { showProductCode: true },
  on(createAction('[Products Page] Toggle Show Product Code'), (state) => ({
    ...
  })) );
);
```



# Products Reducer

```
import { createAction, createReducer, on } from '@ngrx/store';

export const productsReducer = createReducer(
  { showProductCode: true },
  on(createAction('[Products Page] Toggle Show Product Code'), (state) => ({
    ...state,
    showProductCode: !state.showProductCode,
  }))
);
```



# Demo



## Building a reducer to process actions



src &gt; app &gt; products &gt; products-page &gt; A products-page.component.ts &gt; ProductsPageComponent

```
7  @Component({
8    selector: 'app-products-page',
9    templateUrl: './products-page.component.html',
10   styleUrls: ['./products-page.component.css'],
11 })
12 export class ProductsPageComponent {
13   products: Product[] = [];
14   total = 0;
15   loading = true;
16   showProductCode = true;
17   errorMessage = '';
18
19   constructor(private productsService: ProductsService, private store: Store) {
20     this.store.subscribe(store => console.log({ store }));
21   }
22
23   ngOnInit() {
24     this.getProducts();
25   }
26
27   getProducts() {
28     this.productsService.getAll().subscribe({
29       next: (products) => {
30         this.products = products;
31         this.total = sumProducts(products);
32         this.loading = false;
33       },
34       error: (error) => (this.errorMessage = error),
35     });
36   }
}
```

src > app > products > products-page > 5 products-page.component.html > div.card.error-card

Go to component

```
1 <div class="card error-card" *ngIf="errorMessage">
2   Error: {{ errorMessage }}
3 </div>
4
5 <div *ngIf="!loading; else loadingElement">
6   <div class="container">
7     <app-products-list
8       [total]="total"
9       [products]="products"
10      [showProductCode]="showProductCode"
11      (toggleProductCode)="toggleShowProductCode()"
12    ></app-products-list>
13  </div>
14 </div>
15
16 <ng-template #loadingElement>Loading...</ng-template>
17
```

<app-products-list

[total]="total"

[products]="products"

[showProductCode]="showProductCode"

(toggleProductCode)="toggleShowProductCode()"

></app-products-list>

src &gt; app &gt; products &gt; products-page &gt; A products-page.component.ts &gt; ProductsPageComponent

```
7  @Component({
8    selector: 'app-products-page',
9    templateUrl: './products-page.component.html',
10   styleUrls: ['./products-page.component.css'],
11 })
12 export class ProductsPageComponent {
13   products: Product[] = [];
14   total = 0;
15   loading = true;
16   showProductCode = true;
17   errorMessage = '';
18
19   constructor(private productsService: ProductsService, private store: Store) {
20     this.store.subscribe((store) => console.log({ store }));
21   }
22
23   ngOnInit() {
24     this.getProducts();
25   }
26
27   getProducts() {
28     this.productsService.getAll().subscribe({
29       next: (products) => {
30         this.products = products;
31         this.total = sumProducts(products);
32         this.loading = false;
33       },
34       error: (error) => (this.errorMessage = error),
35     });
36   }
}
```

## Demo



**Dispatch an action to change state**



```
src > app > products > products-list > 5 products-list.component.html > ...
  21      <button>
  22          </button>
  23      </div>
  24  </li>
  25  <li>
  26      Total: {{ this.total | currency }}
  27      <div>
  28          <input
  29              id="showProductCode"
  30              type="checkbox"
  31              (change)="toggleProductCode.emit()"
  32              [checked]="showProductCode"
  33          >
  34          <label for="showProductCode">Show Product Code</label>
  35      </div>
  36  </li>
  37 </ul>
  38 </div>
  39
```

```
src > app > products > products-page > 5 products-page.component.html > ...
  Go to component
  1  <div class="card error-card" *ngIf="errorMessage">
  2      Error: {{ errorMessage }}
  3  </div>
  4
  5  <div *ngIf="!loading; else loadingElement">
  6      <div class="container">
  7          <app-products-list
  8              [total]="total"
  9              [products]="products"
 10             [showProductCode]="showProductCode"
 11             (toggleProductCode)="toggleShowProductCode()"
 12         ></app-products-list>
 13     </div>
 14   </div>
 15
 16  <ng-template #loadingElement>Loading...</ng-template>
 17
```