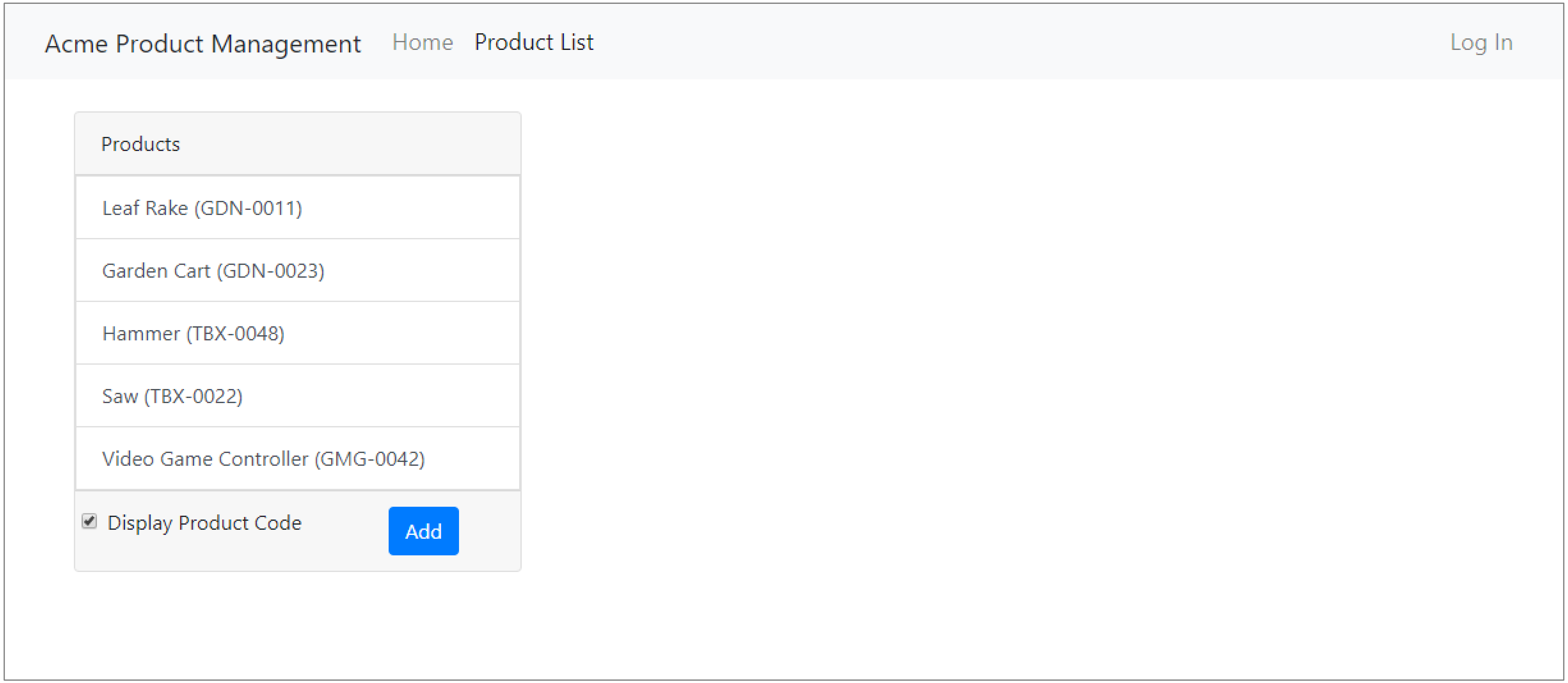
Performing Update Operations

Deborah Kurata
CONSULTANT | SPEAKER | AUTHOR | MVP | GDE

@deborahkurata blogs.msmvps.com/deborahk/

I CAN'~~T~~
DO IT


Acme Product Management Home Product List Log In

Products

Leaf Rake (GDN-0011)

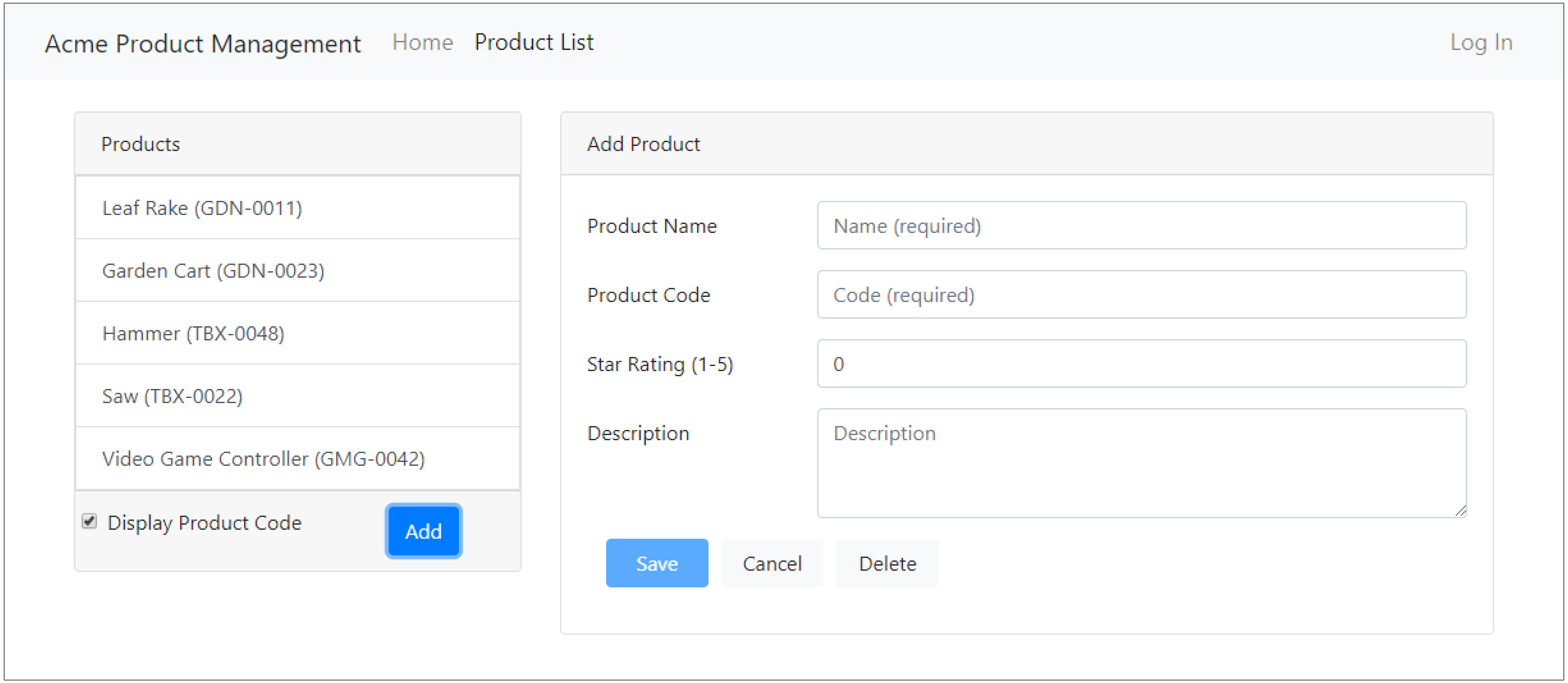
Garden Cart (GDN-0023)

Hammer (TBX-0048)

Saw (TBX-0022)

Video Game Controller (GMG-0042)

☑ Display Product Code



Acme Product Management Home Product List Log In

Products

- Leaf Rake (GDN-0011)
- Garden Cart (GDN-0023)
- Hammer (TBX-0048)
- Saw (TBX-0022)
- Video Game Controller (GMG-0042)

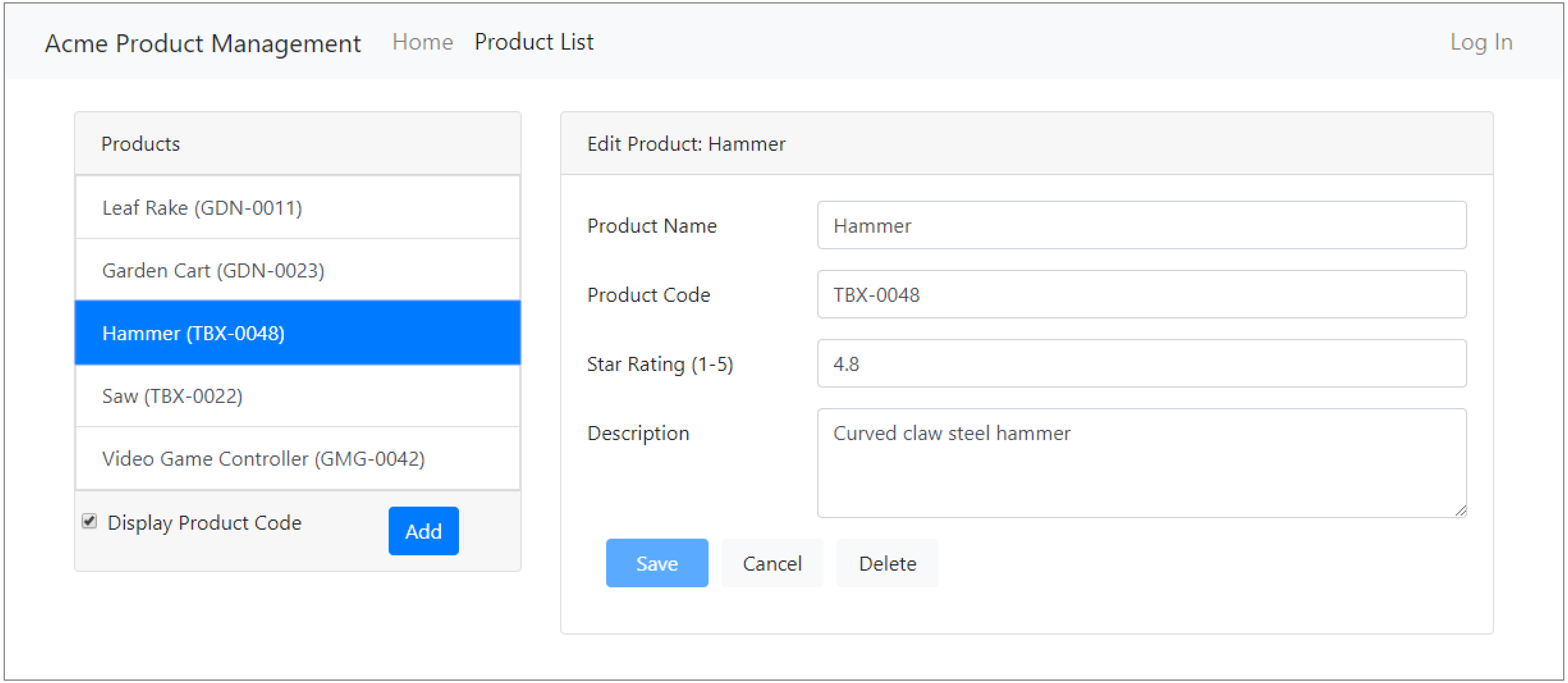
☑ Display Product Code

Add

Add Product

Product Name	Name (required)
Product Code	Code (required)
Star Rating (1-5)	0
Description	Description

Save Cancel Delete



Acme Product Management Home Product List Log In

Products

Leaf Rake (GDN-0011)

Garden Cart (GDN-0023)

Hammer (TBX-0048)

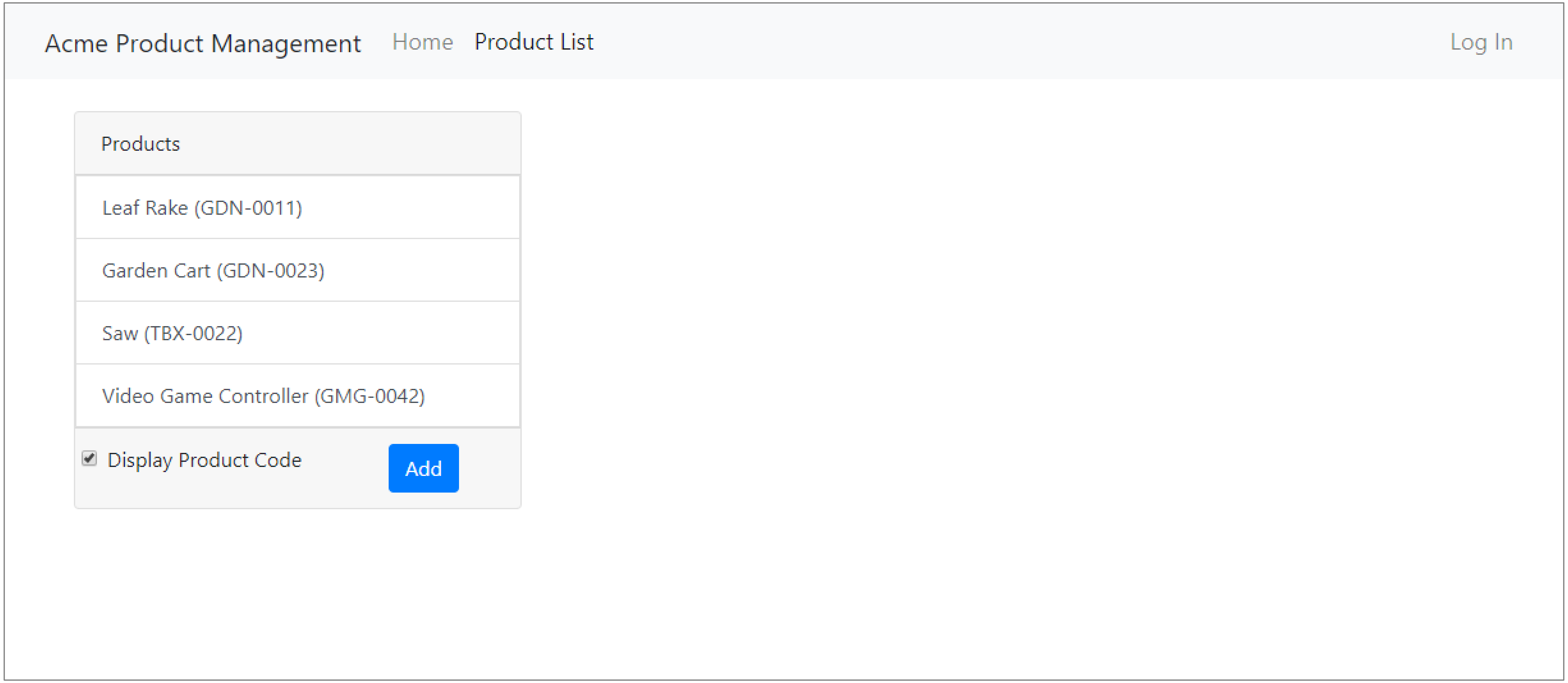
Saw (TBX-0022)

Video Game Controller (GMG-0042)

☑ Display Product Code

Edit Product: Hammer

Product Name	Hammer
Product Code	TBX-0048
Star Rating (1-5)	4.8
Description	Curved claw steel hammer


Acme Product Management Home Product List Log In

Products

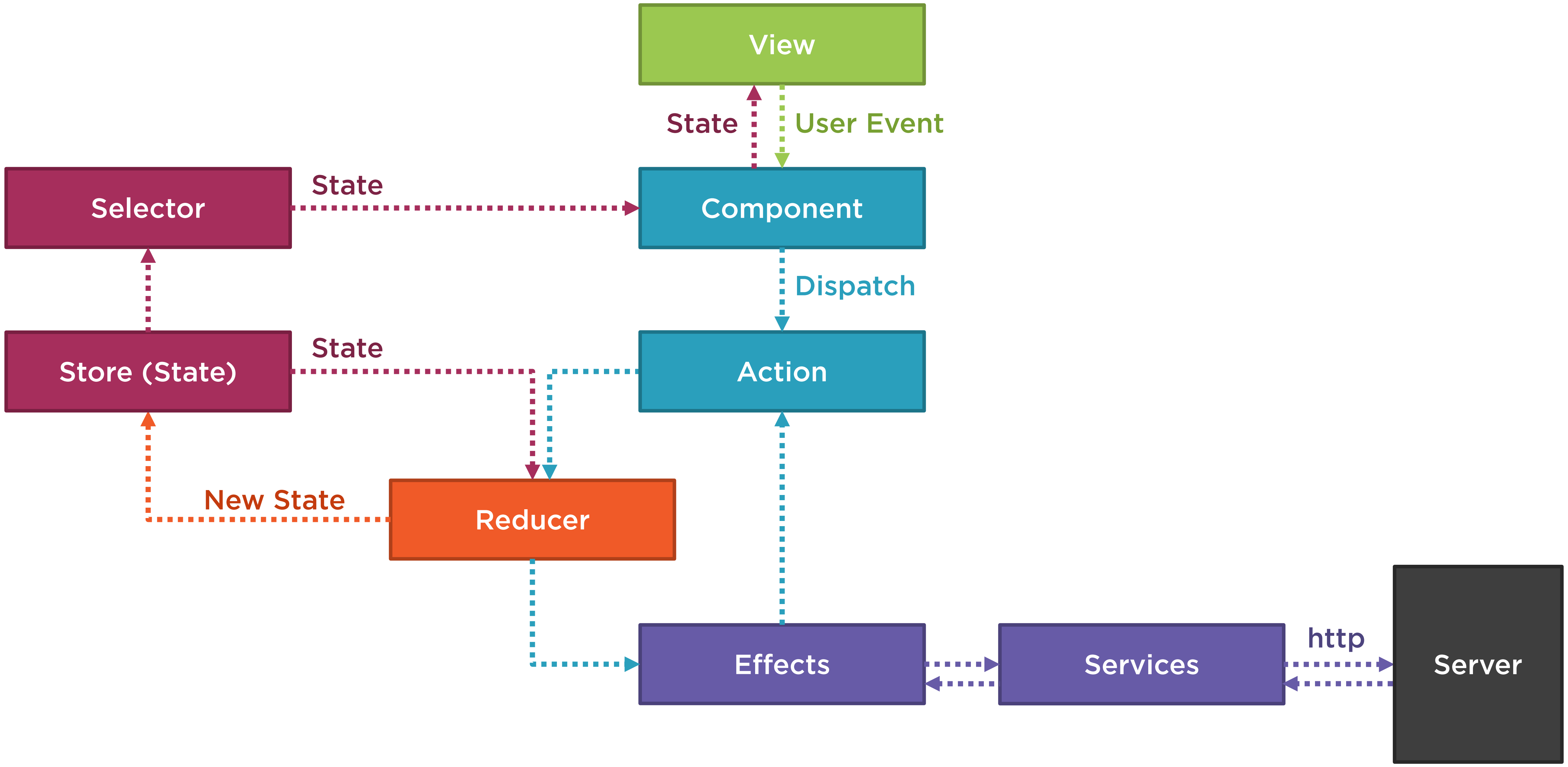
Leaf Rake (GDN-0011)

Garden Cart (GDN-0023)

Saw (TBX-0022)

Video Game Controller (GMG-0042)

[✓] Display Product Code


View
State
User Event
Selector
State
Component
Dispatch
Store (State)
State
Action
New State
Reducer
Effects
Services
http
Server

Module Overview

Identify the state and actions

Strongly type the state and build selectors

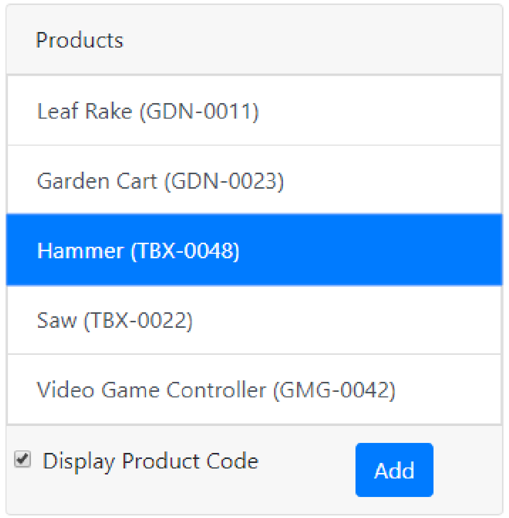
Strongly type the actions with action creators

Dispatch an action

Build the effect to process the action

Process the success and fail actions

Goal: Update a Product


Products

Leaf Rake (GDN-0011)

Garden Cart (GDN-0023)

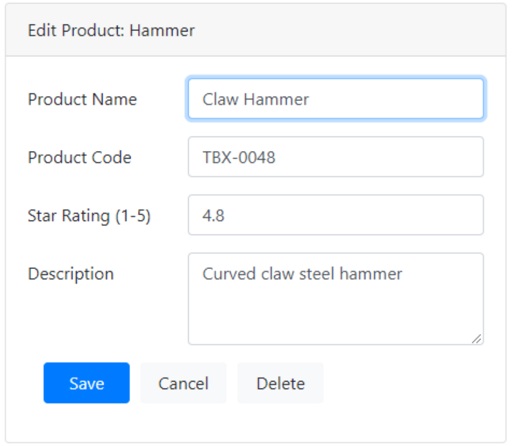
Hammer (TBX-0048)

Saw (TBX-0022)

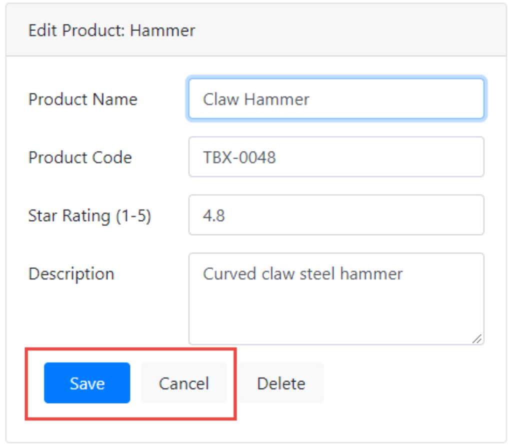
Video Game Controller (GMG-0042)

☑ Display Product Code

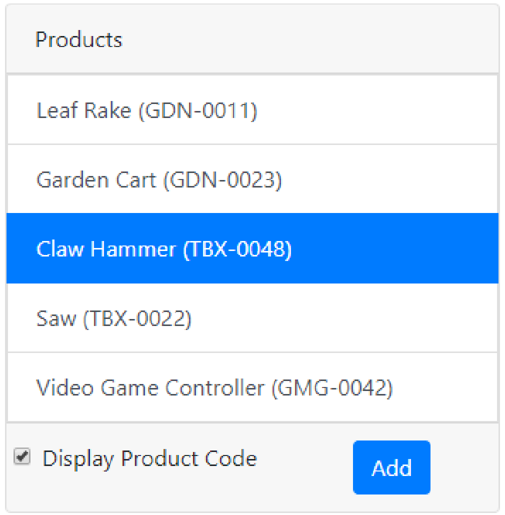
Add


Edit Product: Hammer

Product Name	Claw Hammer
Product Code	TBX-0048
Star Rating (1-5)	4.8
Description	Curved claw steel hammer


Edit Product: Hammer

Product Name	Claw Hammer
Product Code	TBX-0048
Star Rating (1-5)	4.8
Description	Curved claw steel hammer


Products

Leaf Rake (GDN-0011)

Garden Cart (GDN-0023)

Claw Hammer (TBX-0048)

Saw (TBX-0022)

Video Game Controller (GMG-0042)

☑ Display Product Code

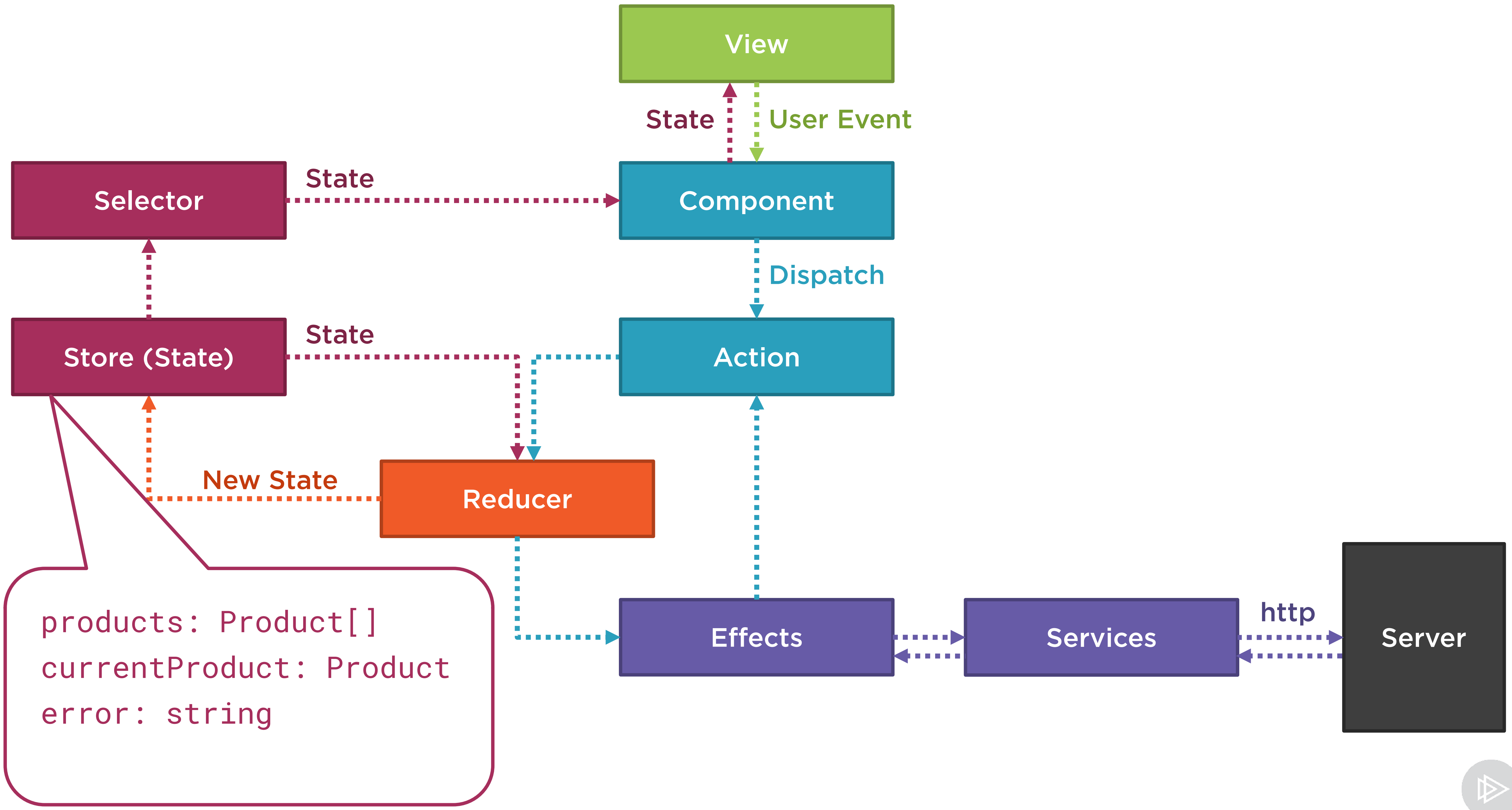
Add

Select a product

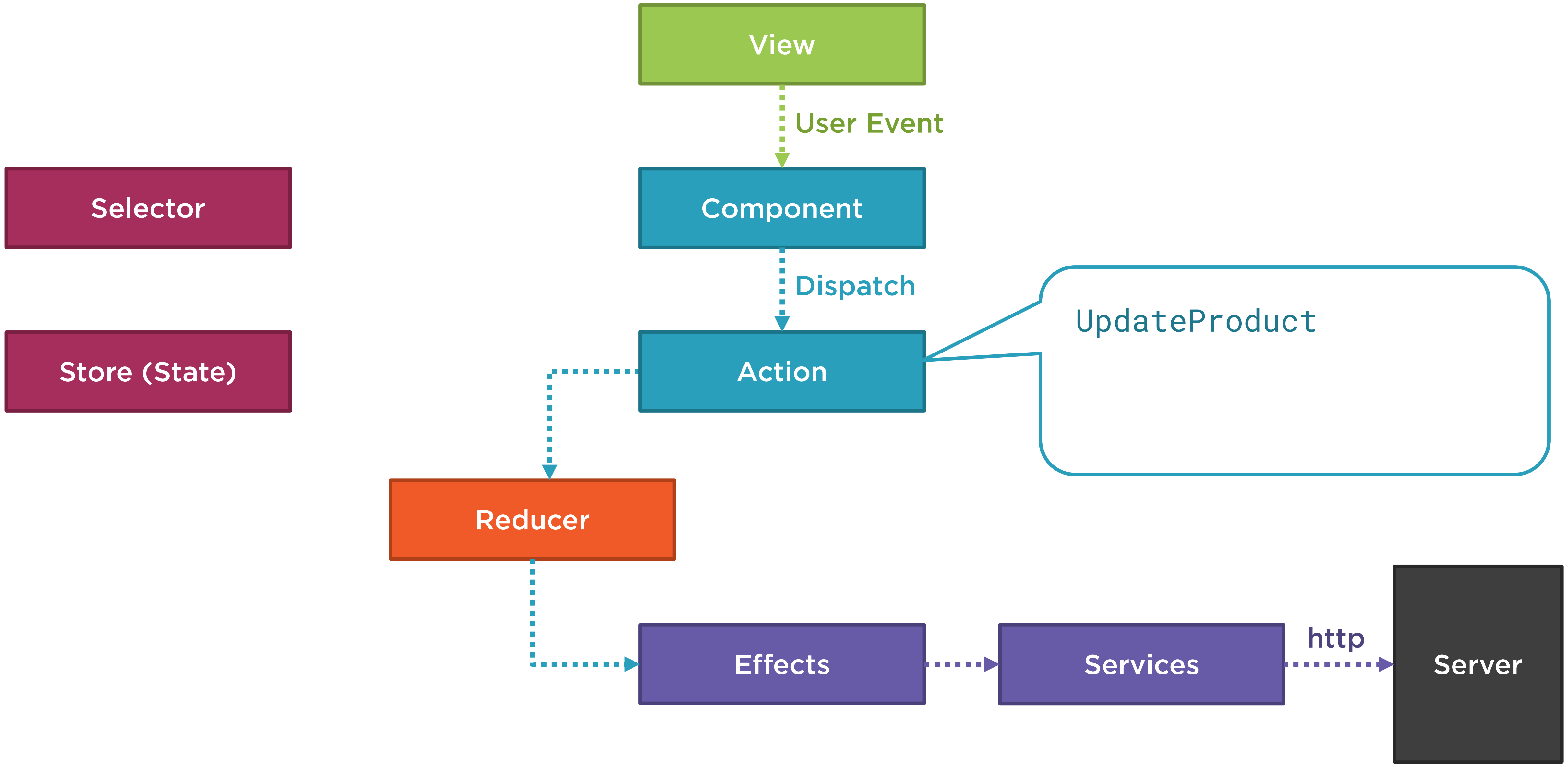
Edit its properties

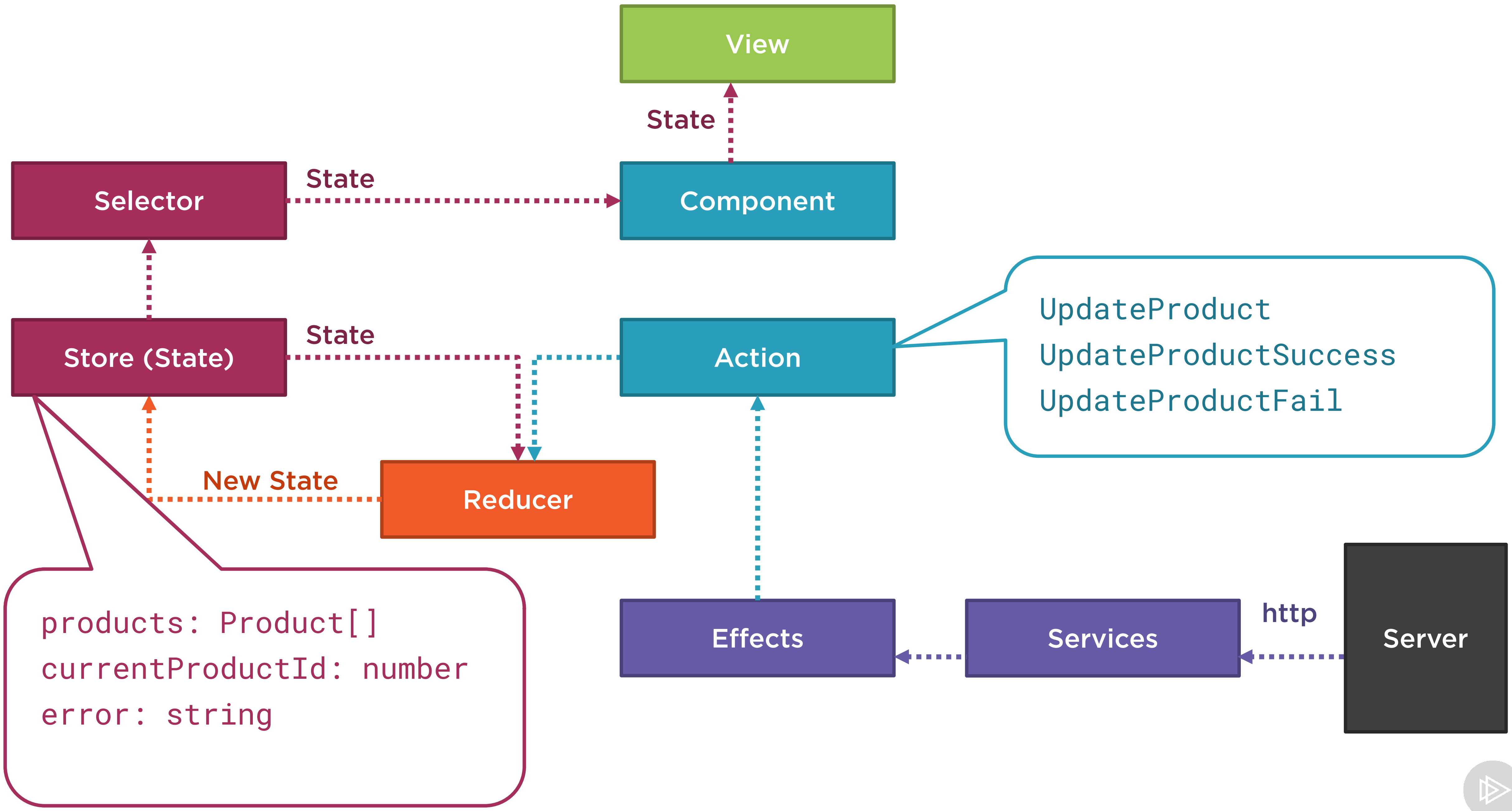
Save or cancel

Display the updated product



```
products: Product[]
currentProduct: Product
error: string
```


View
User Event
Selector
Component
Dispatch
UpdateProduct
Store (State)
Action
Reducer
Effects
Services
http
Server



```
products: Product[]
currentProductId: number
error: string
```

Defining the (Strongly Typed) State

Define an interface

```
export interface ProductState {
 showProductCode: boolean;
 currentProductId: number;
 products: Product[];
}
```

Set initial value

```
const initialState: ProductState = {
 showProductCode: true,
 currentProductId: null,
 products: []
};
```

Build selectors

```
export const getProducts = createSelector(
 getProductFeatureState,
 state => state.products
);
```

Demo

Defining the (strongly typed) state

Defining the (Strongly Typed) Actions

Define action types as named constants

```
export enum ProductActionTypes {
 UpdateProduct = '[Product] Update Product',
 UpdateProductSuccess = '[Product] Update Product Success',
 UpdateProductFail = '[Product] Update Product Fail'
}
```

Build the action creators

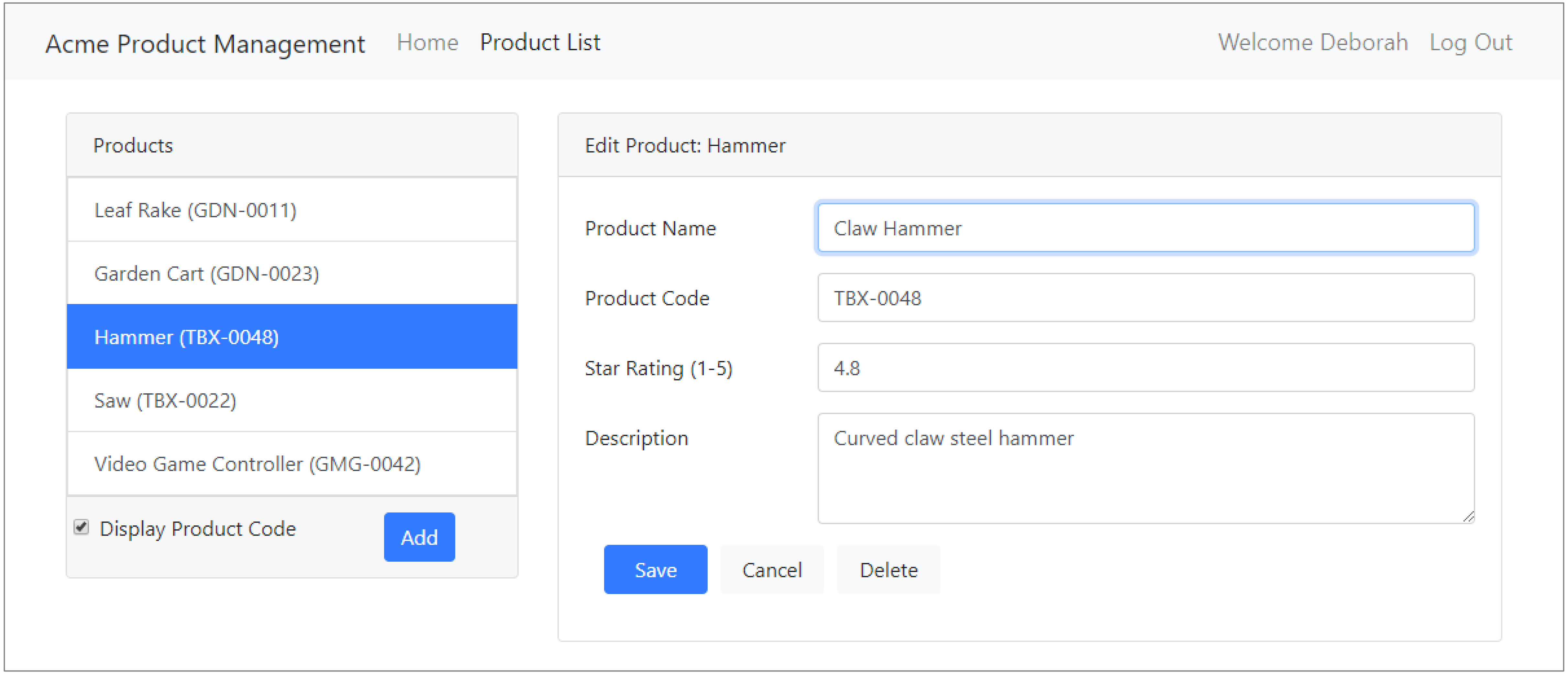
```
export class UpdateProduct implements Action {
 readonly type = ProductActionTypes.UpdateProduct;
 constructor(public payload: Product) {}
}
```

Define a union type

```
export type ProductActions = ToggleProductCode
  | UpdateProduct
  | UpdateProductSuccess
  | UpdateProductFail;
```

Demo

Defining the (strongly typed) actions


Acme Product Management
Home
Product List
Welcome Deborah
Log Out

Products

- Leaf Rake (GDN-0011)
- Garden Cart (GDN-0023)
- Hammer (TBX-0048)
- Saw (TBX-0022)
- Video Game Controller (GMG-0042)

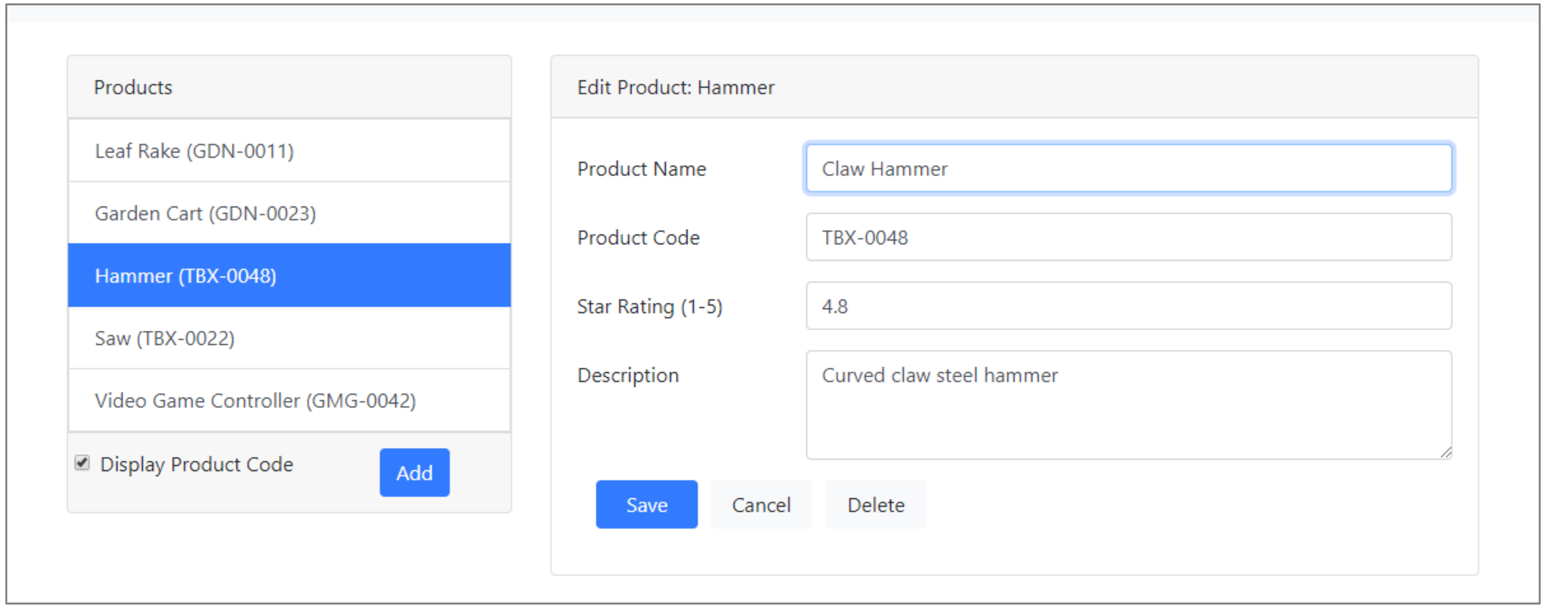
☑ Display Product Code
Add

Edit Product: Hammer

Product Name	Claw Hammer
Product Code	TBX-0048
Star Rating (1-5)	4.8
Description	Curved claw steel hammer

Save
Cancel
Delete

Template-driven Forms


Products

Leaf Rake (GDN-0011)

Garden Cart (GDN-0023)

Hammer (TBX-0048)

Saw (TBX-0022)

Video Game Controller (GMG-0042)

Display Product Code

Edit Product: Hammer

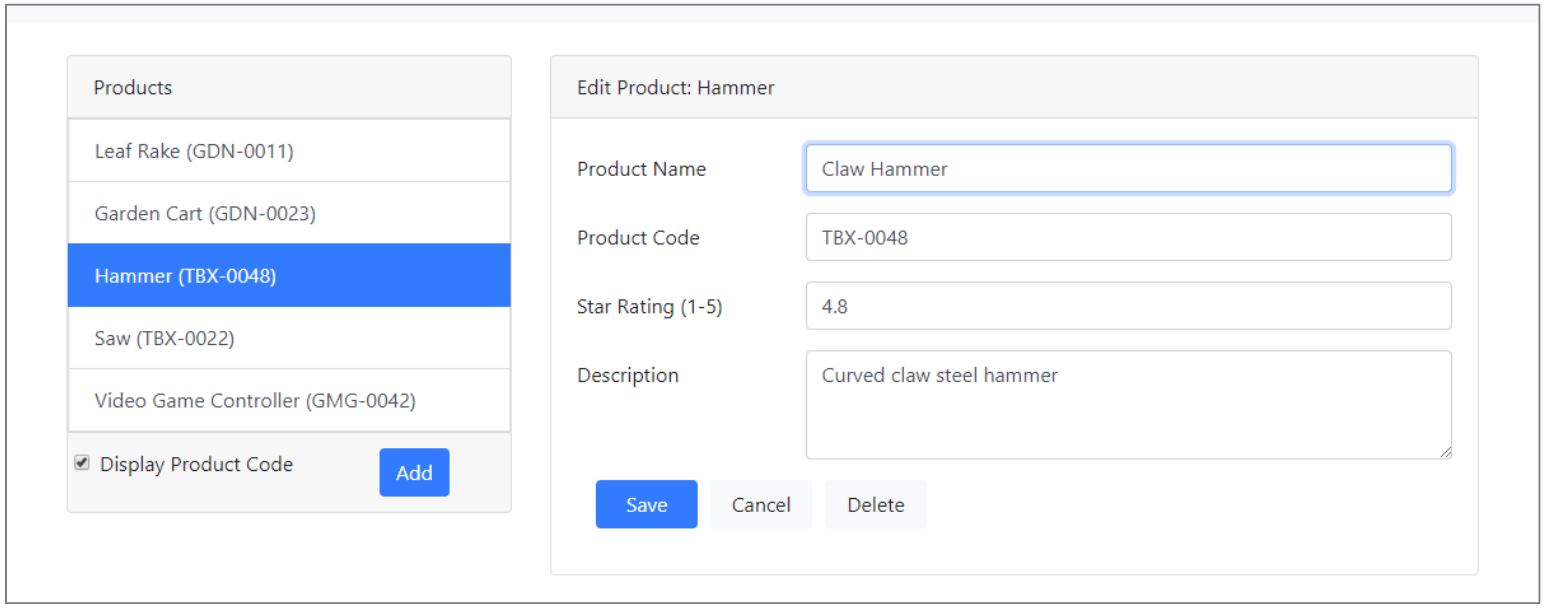
Product Name	Claw Hammer
Product Code	TBX-0048
Star Rating (1-5)	4.8
Description	Curved claw steel hammer

```
<input class="form-control"
       id="productNameId"
       type="text"
       placeholder="Name (required)"
       required
       minlength="3"
       [(ngModel)]="product.productName"
       name="productName"
       #nameVar="ngModel" />
```

```
export class ProductEditComponent {
 pageTitle = 'Product Edit';
 errorMessage = '';

 product: Product;
 ...
}
```

Reactive Forms



Products

Leaf Rake (GDN-0011)

Garden Cart (GDN-0023)

Hammer (TBX-0048)

Saw (TBX-0022)

Video Game Controller (GMG-0042)

☑ Display Product Code

Edit Product: Hammer

Product Name: Claw Hammer

Product Code: TBX-0048

Star Rating (1-5): 4.8

Description: Curved claw steel hammer

```
this.productForm = this.fb.group({
  productName: ['', [Validators.required, Validators.maxLength(50)]],
  productCode: ['', Validators.required],
  starRating: ['', NumberValidators.range(1, 5)],
  description: ''
});
```

Angular Reactive Forms

Dispatching an Action

Inject the store

```
constructor(private store: Store<fromProduct.State>) { }
```

Call the dispatch method

```
this.store.dispatch( ... )
```

Create the action using the action creator

```
import * as productActions from '../state/product.actions';
...
this.store.dispatch(new productActions.UpdateProduct(product));
```

Demo

Dispatching an action

Building the Effect

Build effect service and inject Actions

```
@Injectable()
export class ProductEffects {
 constructor(private actions$: Actions) { }
}
```

Specify a property with the Effect decorator

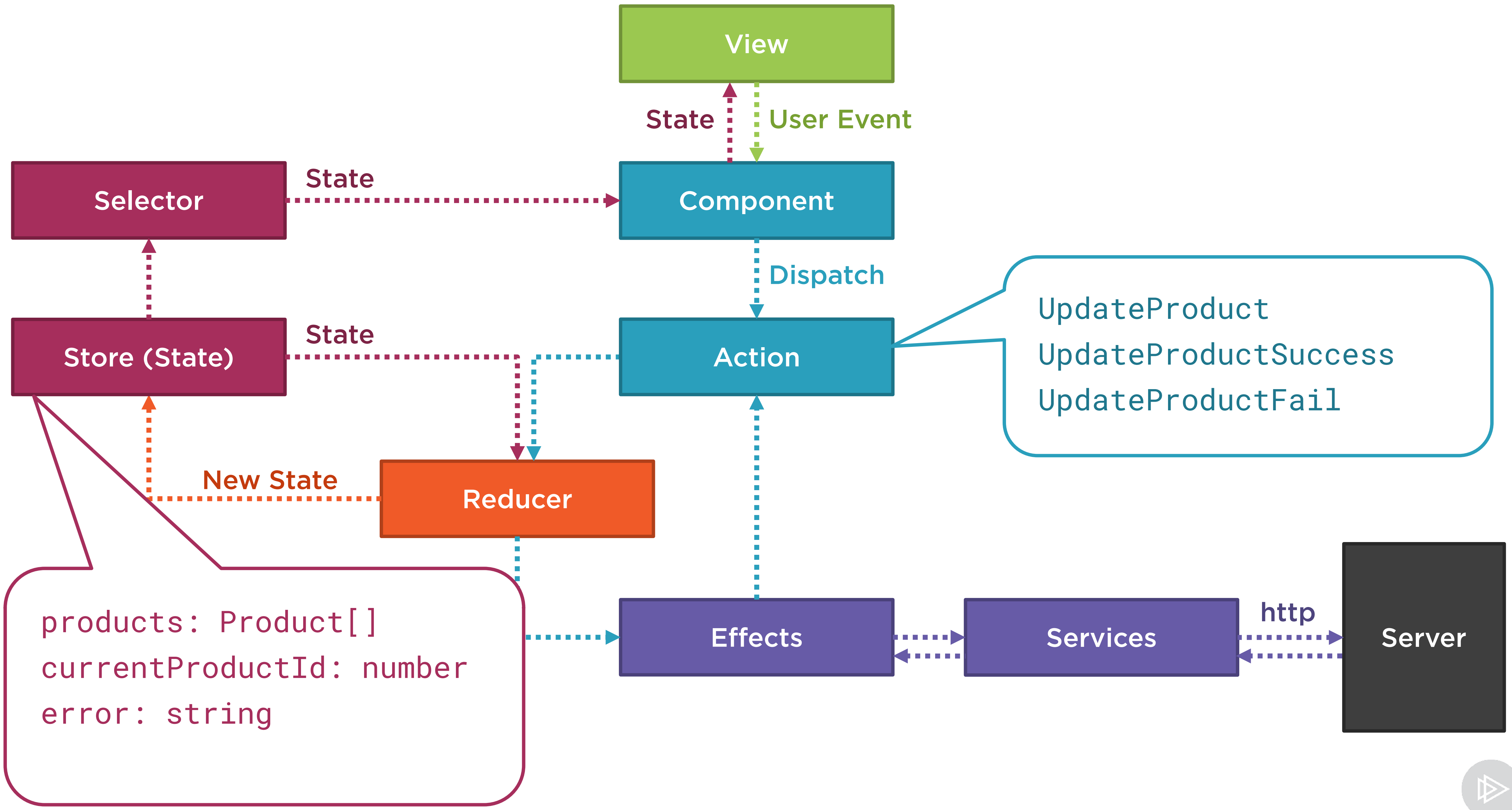
```
@Effect()
updateProduct$: Observable<Action>
```

Build the effect

```
updateProduct$: Observable<Action> = this.actions$.pipe(
 ofType(fromProduct.ProductActionTypes.UpdateProduct),
 map((action: fromProduct.UpdateProduct) => action.payload),
 mergeMap((product: Product) =>
  this.productService.updateProduct(product).pipe(
   map(updatedProduct => (new fromProduct.UpdateProductSuccess(updatedProduct))),
   catchError(err => of(new fromProduct.UpdateProductFail(err)))
)));
```

Demo

Building the effect



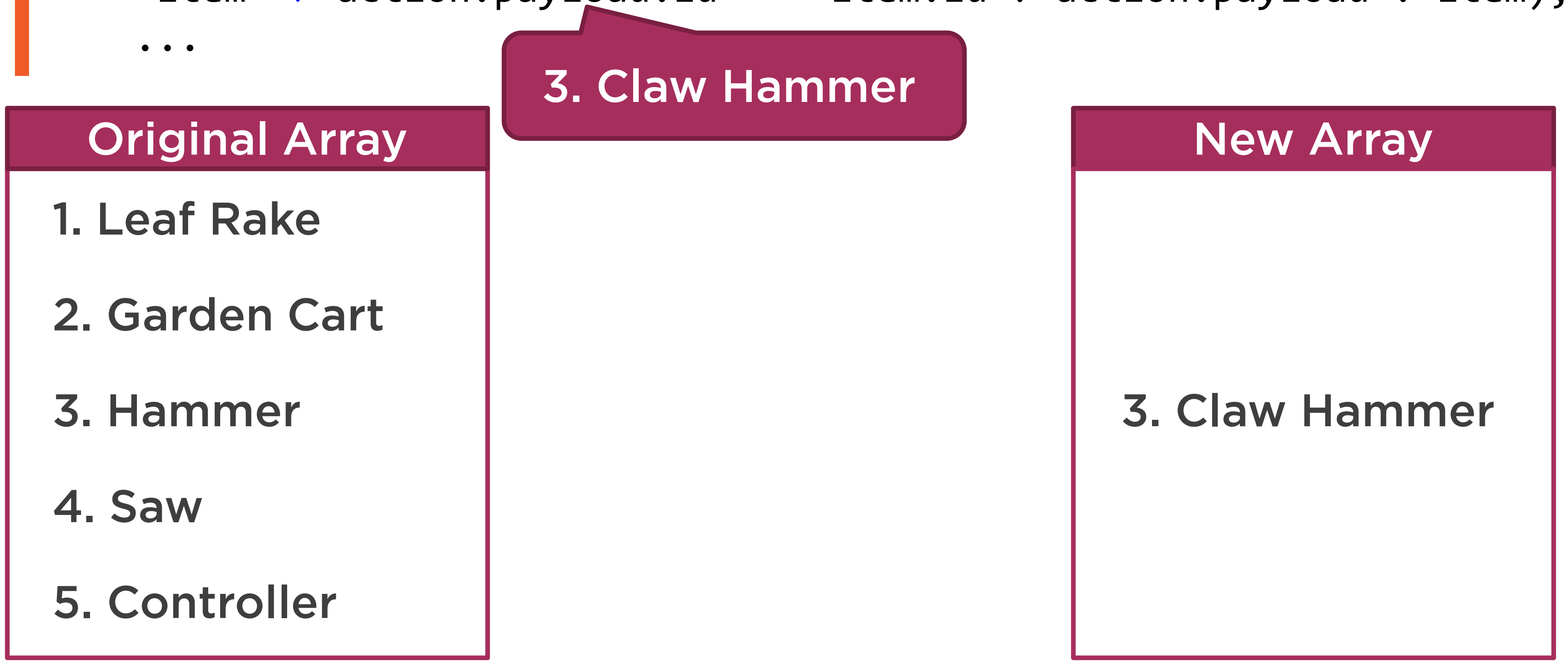
```
UpdateProduct
UpdateProductSuccess
UpdateProductFail
```

```
products: Product[]
currentProductId: number
error: string
```

Processing the Success and Fail Actions

Add a case and build a new array

```
case ProductActionTypes.UpdateProductSuccess:
 const updatedProducts = state.products.map(
  item => action.payload.id === item.id ? action.payload : item);
 ...
```



Processing the Success and Fail Actions

Add a case and build a new array

```
case ProductActionTypes.UpdateProductSuccess:
 const updatedProducts = state.products.map(
  item => action.payload.id === item.id ? action.payload : item);
 ...
```

Return the new state

```
return { ...state, products: updatedProducts,
         currentProductId: action.payload.id,
         error: ''};
```

Add a case and return the error

```
case ProductActionTypes.UpdateProductFail:
  return { ...state, error: action.payload};
```

Immutable vs. Mutable Array Methods

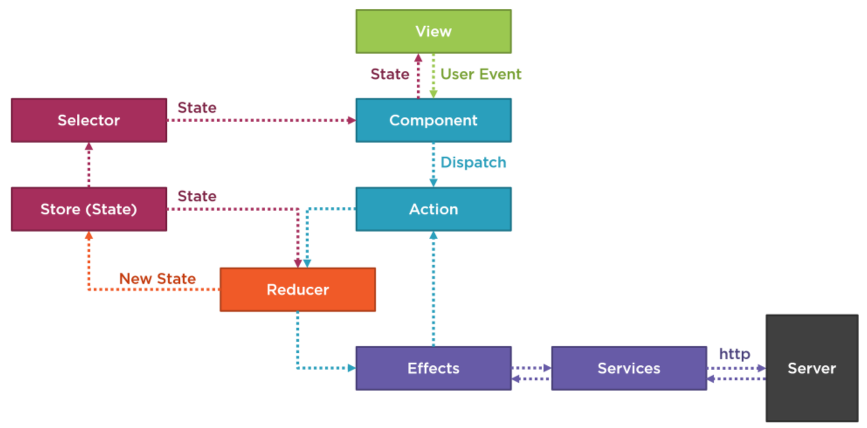
An immutable object or array cannot be modified after it is created.

`state.products.push(action.payload)`	**Mutable**
`state.products.concat(action.payload)`	**Immutable**
`[...state.products, action.payload]`	**Immutable**
`state.products.shift()`	**Mutable**
`state.products.splice(0,2)`	**Mutable**
`state.products.filter(p => p.id !== action.payload.id)`	**Immutable**
`state.products.map(p => p.id === action.payload.id ? action.payload : p)`	**Immutable**
`state.products.forEach(p => p.id === action.payload.id ? action.payload : p)`	**Mutable**

Demo

Processing the success and fail actions

Checklist:
Performing Operations with Side Effects



Identify the state and actions

Strongly type the state and build selectors

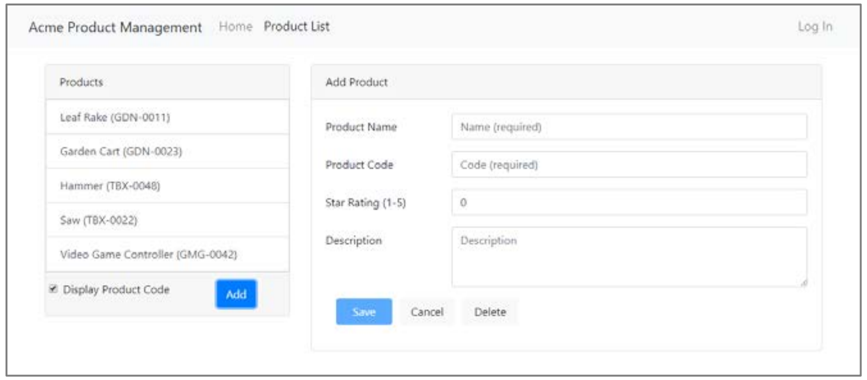
Strongly type the actions with action creators

Dispatch an action to kickoff the operation

Build the effect to perform the operation and dispatch a success or fail action

Process the success and fail actions in the reducer

Homework



Acme Product Management Home Product List Log In

Products

- Leaf Rake (GDN-0011)
- Garden Cart (GDN-0023)
- Hammer (TBX-0048)
- Saw (TBX-0022)
- Video Game Controller (GMG-0042)

☑ Display Product Code Add

Add Product

Product Name	Name (required)
Product Code	Code (required)
Star Rating (1-5)	0
Description	Description

Save Cancel Delete

Identify the state and actions

Define a state interface and selectors

Build action creators

Dispatch an action to kick off the operation

Build the effect to process that action and dispatch the success and fail actions

Process the success and fail actions in the reducer

https://github.com/DeborahK/Angular-NgRx-GettingStarted/tree/master/APM-Demo4