Component Selectors

Introduction

In this document, we'll enhance the previous e-commerce Angular application from **Anatomy of a Component** by incorporating advanced component selectors as described in the https://angular.dev/guide/components/selectors. We'll demonstrate: - Different types of component selectors: element selectors, attribute selectors, class selectors, and pseudo-class selectors. - How to use these selectors in a realistic and complex application. - The implications and use cases for each type of selector.

By the end of this example, you'll have a deeper understanding of how to use various component selectors effectively in an Angular application.

Application Overview

Our enhanced e-commerce application will include the following components: 1) `AppComponent`: The root component. 2) `ProductListComponent`: Displays a list of products. 3) `ProductCardComponent`: Displays individual product details using an **attribute selector**. 4) `SpecialOfferComponent`: Highlights special offers using a **class selector**. 5) `AdminBannerComponent`: Displays an admin banner using a **pseudo-class selector**. 6) `ProductDetailComponent`: Shows detailed information of a selected product using an **element selector**. 7) `ProductService`: Manages product data.

We'll modify existing components and add new ones to demonstrate these selectors in action.

Code

app.component.ts

```
import { Component } from '@angular/core';

@Component({
  selector: 'app-root',
  template: `
    <app-admin-banner></app-admin-banner>
    <h1>E-commerce Store</h1>
    <div class="product-container">
      <app-product-list (selectProduct)="handleSelectProduct($event)"></app-product-
        list>
      <app-product-detail [product]="selectedProduct"></app-product-detail>
    </div>
    <app-cart [cartItems]="cartItems"></app-cart>
  `,
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  cartItems: any[] = [];
  selectedProduct: any = null;

  handleSelectProduct(product: any) {
    this.selectedProduct = product;
  }
}
```

product-list.component.ts

```
import { Component, OnInit, Output, EventEmitter } from '@angular/core';
import { ProductService } from '../product.service';

@Component({
  selector: 'app-product-list',
  template: `
    <h2>Products</h2>
    <app-product-item
      *ngFor="let product of products"
      [product]="product"
      (add)="onAddToCart($event)"
    ></app-product-item>
  `,
  styleUrls: ['./product-list.component.css']
})
export class ProductListComponent implements OnInit {
  products: any[] = [];
  @Output() addToCart = new EventEmitter<any>();

  constructor(private productService: ProductService) {}

  ngOnInit() {
    this.products = this.productService.getProducts();
  }

  onAddToCart(product: any) {
    this.addToCart.emit(product);
  }
}
```

product-list.component.ts

```
import { Component, OnInit, Output, EventEmitter } from '@angular/core';
import { ProductService } from '../product.service';

@Component({
  selector: 'app-product-list',
  template: `
    <h2>Products</h2>
    <div class="product-list">
      <div
        *ngFor="let product of products"
        [product-card]
        [product]="product"
        (add)="onAddToCart($event)"
        (viewDetails)="onViewDetails($event)"
      ></div>
    </div>
  `,
  styleUrls: ['./product-list.component.css']
})
export class ProductListComponent implements OnInit {
  products: any[] = [];
  @Output() addToCart = new EventEmitter<any>();
  @Output() selectProduct = new EventEmitter<any>();

  constructor(private productService: ProductService) {}

  ngOnInit() {
```

```
    this.products = this.productService.getProducts();
  }

  onAddToCart(product: any) {
    this.addToCart.emit(product);
  }

  onViewDetails(product: any) {
    this.selectProduct.emit(product);
  }
}
```

product-card.component.ts

```
import { Component, Input, Output, EventEmitter } from '@angular/core';

@Component({
  selector: '[product-card]',
  template: `
    <div
      [class.special-offer]="product.isSpecialOffer"
      (click)="viewDetails()"
    >
      <h3>{{ product.name }}</h3>
      <p>{{ product.description }}</p>
      <ng-content></ng-content>
      <button (click)="addToCart($event)">Add to Cart</button>
    </div>
  `,
  styleUrls: ['./product-card.component.css']
})
export class ProductCardComponent {
  @Input() product: any;
  @Output() add = new EventEmitter<any>();
  @Output() viewDetails = new EventEmitter<any>();

  addToCart(event: Event) {
    event.stopPropagation();
    this.add.emit(this.product);
  }

  viewDetails() {
    this.viewDetails.emit(this.product);
  }
}
```

special-offer.component.ts

```
import { Component } from '@angular/core';

@Component({
  selector: '.special-offer',
  template: `
    <div class="special-offer-banner">
      <p> Special Offer! </p>
    </div>
    <ng-content></ng-content>
  `,
  styleUrls: ['./special-offer.component.css']
})
```

```
export class SpecialOfferComponent {}
```

admin-banner.component.ts

```
import { Component } from '@angular/core';

@Component({
  selector: 'app-admin-banner::ng-deep',
  template: `
    <div class="admin-banner">
      <p>Admin Mode Activated</p>
    </div>
  `,
  styleUrls: ['./admin-banner.component.css']
})
export class AdminBannerComponent {}
```

product-detail.component.ts

```
import { Component, Input } from '@angular/core';

@Component({
  selector: 'app-product-detail',
  template: `
    <div *ngIf="product" class="product-detail">
      <h2>Product Details</h2>
      <h3>{{ product.name }}</h3>
      <p>{{ product.description }}</p>
      <p><strong>Price:</strong> {{ product.price | currency }}</p>
      <p><strong>Specifications:</strong></p>
      <ul>
        <li *ngFor="let spec of product.specifications">{{ spec }}</li>
      </ul>
      <button (click)="addToCart()">Add to Cart</button>
    </div>
  `,
  styleUrls: ['./product-detail.component.css']
})
export class ProductDetailComponent {
  @Input() product: any;
  @Output() add = new EventEmitter<any>();

  addToCart() {
    this.add.emit(this.product);
  }
}
```

cart.component.ts

```
import { Component, Input } from '@angular/core';

@Component({
  selector: 'app-cart',
  template: `
    <h2>Shopping Cart</h2>
    <div *ngFor="let item of cartItems">
      {{ item.name }} - {{ item.price | currency }}
    </div>
```

```
    <p *ngIf="cartItems.length === 0">Your cart is empty.</p>
  `,
  styleUrls: ['./cart.component.css']
})
export class CartComponent {
  @Input() cartItems: any[] = [];
}
```

product.service.ts

```
import { Injectable } from '@angular/core';

@Injectable({
  providedIn: 'root'
})
export class ProductService {
  private products = [
    {
      id: 1,
      name: 'Laptop',
      description: 'A high-performance laptop',
      price: 1299.99,
      isSpecialOffer: true,
      specifications: ['16GB RAM', '512GB SSD', 'Intel i7 Processor']
    },
    {
      id: 2,
      name: 'Smartphone',
      description: 'A powerful smartphone',
      price: 799.99,
      isSpecialOffer: false,
      specifications: ['128GB Storage', '6GB RAM', 'OLED Display']
    },
    {
      id: 3,
      name: 'Headphones',
      description: 'Noise-cancelling headphones',
      price: 199.99,
      isSpecialOffer: true,
      specifications: ['Wireless', '20h Battery Life', 'Bluetooth 5.0']
    }
  ];

  getProducts() {
    return this.products;
  }
}
```

app.module.ts

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';

import { AppComponent } from './app.component';
import { ProductListComponent } from './product-list/product-list.component';
import { ProductCardComponent } from './product-card.component';
import { CartComponent } from './cart/cart.component';
import { SpecialOfferComponent } from './special-offer.component';
import { AdminBannerComponent } from './admin-banner.component';
import { ProductDetailComponent } from './product-detail.component';
```

```
@NgModule({
  declarations: [
    AppComponent,
    ProductListComponent,
    ProductCardComponent,
    CartComponent,
    SpecialOfferComponent,
    AdminBannerComponent,
    ProductDetailComponent
  ],
  imports: [BrowserModule],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule {}
```

Styles and Templates

For brevity, the styles (`.css` files) are minimal. You can style the components as needed.

Explanation of the Code

AppComponent

- **Purpose**: Acts as the root component of the application.
- **Template**:
  - Includes the `<app-admin-banner>` component, which uses a pseudo-class selector.
  - Displays the application title.
  - Contains a `product-container` div that holds the `<app-product-list>` and `<app-product-detail>` components.
  - Includes the `<app-cart>` component to display the shopping cart.
- **Logic**:
  - Maintains `cartItems` to track items added to the cart.
  - Maintains `selectedProduct` to track the currently selected product.
  - Implements `handleSelectProduct()` to update the selected product when a product is clicked.

ProductListComponent

- **Purpose**: Displays a list of products fetched from the `ProductService`.
- **Template**:
  - Iterates over `products` using `*ngFor`.
  - Uses the `[product-card]` attribute selector to render each product.
  - Binds `[product]` input and `(add)` and `(viewDetails)` outputs to handle events.
- **Logic**:
  - Fetches products on initialization.
  - Emits `addToCart` and `selectProduct` events when corresponding actions occur.

ProductCardComponent

- **Purpose**: Displays individual product details.
- **Selector**: Uses an **attribute selector** `[product-card]`.
- **Template**:
  - Applies the `special-offer` CSS class dynamically if `product.isSpecialOffer` is true.

    - Displays product name and description.
    - Includes an “Add to Cart” button.
- **Logic**:
    - Uses `@Input()` to accept a `product`.
    - Emits `add` and `viewDetails` events.
    - Prevents click event propagation when adding to cart to avoid triggering `viewDetails`.

SpecialOfferComponent

- **Purpose**: Highlights special offer products.
- **Selector**: Uses a **class selector** `.special-offer`.
- **Template**:
    - Displays a special offer banner.
    - Projects any content within the component using `<ng-content>`.
- **Usage**:
    - Automatically applied when the `special-offer` CSS class is present on an element.

AdminBannerComponent

- **Purpose**: Displays an admin banner.
- **Selector**: Uses a **pseudo-class selector** `app-admin-banner::ng-deep`.
- **Template**:
    - Displays a banner indicating admin mode.
- **Note**:
    - The `::ng-deep` pseudo-class allows styling child components from the parent.

ProductDetailComponent

- **Purpose**: Shows detailed information about a selected product.
- **Template**:
    - Displays product details only if a product is selected.
    - Shows specifications of the product in a list.
    - Includes an “Add to Cart” button.
- **Logic**:
    - Accepts `product` as an input.
    - Emits add `event` when the product is added to the cart.

CartComponent

- **Purpose**: Displays the items in the shopping cart.
- **Template**:
    - Iterates over `cartItems` to display each item’s name and price.
    - Shows a message if the cart is empty.
- **Logic**:
    - Accepts `cartItems` as an input.

ProductService

- **Purpose**: Provides product data to components.
- **Logic**:
    - Contains a list of products with additional properties like `isSpecialOffer` and `specifications`.
    - Provides `getProducts()` method to retrieve the product list.

AppModule

- **Purpose**: The root module that bootstraps the application.
- **Declarations**:
  - Lists all components used in the application, including new components like `ProductCardComponent`, `SpecialOfferComponent`, `AdminBannerComponent`, and `ProductDetailComponent`.
- **Imports**:
  - `BrowserModule` is imported to run the app in a browser.
- **Bootstrap**:
  - Bootstraps the `AppComponent` to launch the application.

Key Angular Concepts Demonstrated

Component Selectors

- **Element Selector**: Used in `AppComponent` and `ProductDetailComponent` to define components as HTML elements.
- **Attribute Selector**: Used in `ProductCardComponent` to apply the component to any element with the `product-card` attribute.
- **Class Selector**: Used in `SpecialOfferComponent` to apply the component to any element with the `special-offer` class.
- **Pseudo-Class Selector**: Used in `AdminBannerComponent` to apply styles deeply within the component tree.

Component Interaction

- **Inputs and Outputs**: Components communicate via `@Input()` and `@Output()` decorators to pass data and events.
- **Event Handling**: Components emit events to notify parent components of actions, such as adding a product to the cart or selecting a product for details.

View Encapsulation and Content Projection

- **View Encapsulation**: Styles are encapsulated within components, with `::ng-deep` allowing styles to penetrate component boundaries.
- **Content Projection**: `<ng-content>` is used in `SpecialOfferComponent` to allow content to be projected into the component.

Services and Dependency Injection

- **ProductService**: Provides product data and is injected into components to supply data.

Pipes

- **Currency Pipe**: Used in `CartComponent` and `ProductDetailComponent` to format product prices.

Conclusion

This enhanced example builds upon the previous e-commerce application by incorporating advanced component selectors. It demonstrates how to use element, attribute, class, and pseudo-class selectors effectively in an Angular application. The example also reinforces key Angular concepts such as component interaction, view encapsulation, and dependency injection.