

Section 20 Challenge Solution

1. Create the components, service and module

```
ng g m orders
ng g m orders/orders-routing --flat
ng g c orders/orders --flat --skip-tests
ng g s orders/orders --skip-tests
ng g c order-detailed --skip-tests
```

2. Add the routes for the orders

```
import { NgModule } from '@angular/core';
import { CommonModule } from '@angular/common';
import { RouterModule, Routes } from '@angular/router';
import { OrdersComponent } from './orders.component';
import { OrderDetailedComponent } from './order-detailed/order-detailed.component';

const routes: Routes = [
  {path: '', component: OrdersComponent},
  {path: ':id', component: OrderDetailedComponent, data: {breadcrumb: {alias: 'OrderDetailed'}}}
]

@NgModule({
  declarations: [],
  imports: [
    RouterModule.forChild(routes)
  ],
  exports: [RouterModule]
})
export class OrdersRoutingModule { }
```

```
import { NgModule } from '@angular/core';
import { CommonModule } from '@angular/common';
import { OrdersComponent } from './orders.component';
import { OrderDetailedComponent } from './order-detailed/order-detailed.component';
import { OrdersRoutingModule } from './orders-routing.module';

@NgModule({
  declarations: [OrdersComponent, OrderDetailedComponent],
  imports: [
    CommonModule,
    OrdersRoutingModule
  ]
})
export class OrdersModule { }
```

3. Lazy load the orders module in the app-routing.module.ts

```
import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';
import { AuthGuard } from '../core/guards/auth.guard';
import { NotFoundComponent } from '../core/not-found/not-found.component';
import { ServerErrorComponent } from '../core/server-error/server-error.component';
import { TestErrorComponent } from '../core/test-error/test-error.component';
import { HomeComponent } from '../home/home.component';

const routes: Routes = [
  // omitted
  {
    path: 'orders',
    canActivate: [AuthGuard],
    loadChildren: () => import('../orders/orders.module').then(mod => mod.OrdersModule),
    data: { breadcrumb: 'Orders' }
  },
  {
    path: 'account', loadChildren: () => import('../account/account.module').then(mod => mod.AccountModule),
    data: { breadcrumb: {skip: true} }
  },
  { path: '**', redirectTo: 'not-found', pathMatch: 'full' }
];

@NgModule({
  imports: [RouterModule.forRoot(routes)],
  exports: [RouterModule]
})
export class AppRoutingModule { }
```

4. Add 2 methods in the order service to get the list of orders and a single order

```
import { HttpClient } from '@angular/common/http';
import { Injectable } from '@angular/core';
import { environment } from 'src/environments/environment';
import { Order } from '../shared/models/order';

@Injectable({
  providedIn: 'root'
})
export class OrdersService {
  baseUrl = environment.apiUrl;

  constructor(private http: HttpClient) { }

  getOrdersForUser() {
    return this.http.get<Order[]>(this.baseUrl + 'orders');
  }

  getOrderDetailed(id: number) { }
```

```

    return this.http.get<Order>(this.baseUrl + 'orders/' + id);
  }
}

```

5. Create the orders list in the orders.component.ts and its template (HTML Template available in the snippets provided in section20 folder in StudentAssets)

```

import { Component, OnInit } from '@angular/core';
import { Order } from '../shared/models/order';
import { OrdersService } from './orders.service';

@Component({
  selector: 'app-orders',
  templateUrl: './orders.component.html',
  styleUrls: ['./orders.component.scss']
})
export class OrdersComponent implements OnInit {
  orders: Order[] = [];

  constructor(private orderService: OrdersService) { }

  ngOnInit(): void {
    this.getOrders();
  }

  getOrders() {
    this.orderService.getOrdersForUser().subscribe({
      next: orders => this.orders = orders
    })
  }
}

```

```

<div class="container mt-5">
  <div class="row">
    <div class="col-12">
      <table class="table table-hover" style="cursor: pointer;">
        <thead class="thead-light">
          <tr>
            <th>Order</th>
            <th>Date</th>
            <th>Total</th>
            <th>Status</th>
          </tr>
        </thead>
        <tbody>
          <tr *ngFor="let order of orders" routerLink="/orders/{{order.id}}">
            <th># {{order.id}}</th>
            <td>{{order.orderDate | date: 'medium'}}</td>

```

```

                <td>{{order.total | currency}}</td>
                <td>{{order.status}}</td>
            </tr>
        </tbody>
    </table>
</div>
</div>
</div>

```

6. In the order-detailed component get the individual order and display it in the browser. The HTML has been provided for this in the Section20 folder in the StudentAssets folder.

```

import { Component, OnInit } from '@angular/core';
import { ActivatedRoute, Router } from '@angular/router';
import { Order } from 'src/app/shared/models/order';
import { BreadcrumbService } from 'xng-breadcrumb';
import { OrdersService } from '../orders.service';

@Component({
  selector: 'app-order-detailed',
  templateUrl: './order-detailed.component.html',
  styleUrls: ['./order-detailed.component.scss']
})
export class OrderDetailedComponent implements OnInit {
  order?: Order;

  constructor(private orderService: OrdersService, private route: ActivatedRoute,
    private bcService: BreadcrumbService) {}

  ngOnInit(): void {
    const id = this.route.snapshot.paramMap.get('id');
    id && this.orderService.getOrderDetailed(+id).subscribe({
      next: order => {
        this.order = order;
        this.bcService.set('@OrderDetailed', `Order# ${order.id} - ${order.status}`);
      }
    })
  }
}

```

```

<div class="container mt-5">
  <div class="row" *ngIf="order">
    <div class="col-8">
      <div>
        <div class="table-responsive">
          <table class="table">
            <thead>
              <tr>
                <th scope="col" class="border-0 bg-light">

```

```

        <div class="p-2 text-uppercase">Product</div>
      </th>
      <th scope="col" class="border-0 bg-light">
        <div class="py-2 text-uppercase">Price</div>
      </th>
      <th scope="col" class="border-0 bg-light">
        <div class="py-2 text-uppercase">Quantity</div>
      </th>
      <th scope="col" class="border-0 bg-light">
        <div class="py-2 text-uppercase">Total</div>
      </th>
    </tr>
  </thead>
  <tbody>
    <tr *ngFor="let item of order.orderItems">
      <th scope="row">
        <div class="p-2">
          
          <div class="ml-3 d-inline-block align-middle">
            <h5 class="mb-0">
              {{item.productName}}
            </h5>
          </div>
        </div>
      </th>
      <td class="align-middle"><strong>{{item.price | currency}}</strong></td>
      <td class="align-middle">
        <span class="font-weight-bold px-2">{{item.quantity}}</span>
      </td>
      <td class="align-middle"><strong>{{item.price * item.quantity | currency}}</strong></td>
    </tr>
  </tbody>
</table>
</div>
</div>
</div>
<div class="col-4">
  <div class="bg-light px-4 py-3 text-uppercase font-weight-bold">Order summary</div>
  <div class="p-4">
    <ul class="list-unstyled mb-1">
      <li class="d-flex justify-content-between py-3 border-bottom">
        <strong class="text-muted">Order subtotal</strong>
        <strong>{{order.subtotal | currency}}</strong>
      </li>
      <li class="d-flex justify-content-between py-3 border-bottom">
        <strong class="text-muted">Shipping and handling</strong>
        <strong>{{order.shippingPrice | currency}}</strong>
      </li>
      <li class="d-flex justify-content-between py-3 border-bottom">
        <strong class="text-muted">Total</strong>
        <strong>{{order.total | currency}}</strong>
      </li>
    </ul>
  </div>
</div>

```

```

    </div>
  </div>
</div>

```

- Update the checkout-success page to allow the user to view the created order or go to the orders list, depending on whether or not we have the order. We need to get the order id from NavigationExtras.

```

import { Component } from '@angular/core';
import { Router } from '@angular/router';
import { Order } from 'src/app/shared/models/order';

@Component({
  selector: 'app-checkout-success',
  templateUrl: './checkout-success.component.html',
  styleUrls: ['./checkout-success.component.scss']
})
export class CheckoutSuccessComponent {
  order?: Order

  constructor(private router: Router) {
    const navigation = this.router.getCurrentNavigation();
    this.order = navigation?.extras?.state as Order
  }
}

```

```

<div class="container mt-5">
  <div><i class="fa fa-check-circle fa-5x" style="color: green;"></i></div>
  <h2>Thank you. Your order is confirmed</h2>
  <p class="mb-4">Your order has not shipped yet, but this is to be expected as we are not a real store!</p>
  <button *ngIf="order"
    routerLink="/orders/{{order.id}}"
    class="btn btn-outline-success">
    View your order
  </button>
  <button *ngIf="!order"
    routerLink="/orders" class="btn btn-outline-success">
    View your orders
  </button>
</div>

```