Routing Steps :

ng new RouteDemo

ng g c home

ng g c about

ng g c contact

ng g c product

ng g c error

ng g class product

product.ts

export class Product {

    constructor(pid:number, pname:string,price:number)

    {

        this.productId=pid;

        this.name=pname;

        this.price=price;

    }

    productId:number;

    name:string;

    price :number;

}

Ng g s product

Product.service.ts

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

import { Product } from './product';

@Injectable({

  providedIn: 'root'

})

export class ProductService {

  public getProducts(){

    let products:Product[];

    products=[

      new Product(1,'mac book',150000),

      new Product(2,'ipad air',60000),

      new Product(3,'iphone',139000)

    ]

    return products;

  }

  constructor() { }

}

[home.component.html](http://home.component.html)

<h1>Welcome!</h1>

<p>This is Home Component </p>

Contact.component.html

<h1>Contact Us</h1>

<p> Corporate Training and Consulting </p>

Product.component.ts

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

import { Product } from '../product';

import { ProductService } from '../product.service';

@Component({

  selector: 'app-product',

  templateUrl: './product.component.html',

  styleUrls: ['./product.component.css']

})

export class ProductComponent implements OnInit {

  products:Product[]=[];

  constructor(private productService:ProductService){

  }

  ngOnInit() {

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

  }

}

Product.component.html

<p>product works!</p>

<h1>Product List</h1>

<div class='table-responsive'>

    <table class='table'>

        <thead>

            <tr>

                <th>ID</th>

                <th>Name</th>

                <th>Price</th>

            </tr>

        </thead>

        <tbody>

            <tr \*ngFor="let product of products;">

                <td>{{product.productId}}</td>

                <td>

                    <a [routerLink]="['detail',product.productId]">

                        {{product.name}}

                    </a>

                </td>

                <td>{{product.price}}</td>

            </tr>

        </tbody>

    </table>

</div>

<router-outlet></router-outlet>

Product.service.ts

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

import { Product } from './product';

@Injectable({

  providedIn: 'root'

})

export class ProductService {

  public getProducts(){

    let products:Product[];

    products=[

      new Product(1,'mac book',150000),

      new Product(2,'ipad air',60000),

      new Product(3,'iphone',139000)

    ]

    return products;

  }

  public getProduct(id:any) {

    let products:Product[]=this.getProducts();

    return products.find(p => p.productId==id);

}

  constructor() { }

}

app-routing.module.ts

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

import { RouterModule, Routes } from '@angular/router';

import { AboutComponent } from './about/about.component';

import { ContactComponent } from './contact/contact.component';

import { ErrorComponent } from './error/error.component';

import { HomeComponent } from './home/home.component';

import { ProductComponent } from './product/product.component';

const routes: Routes = [

  {path:'home', component:HomeComponent},

  {path:'about',component:AboutComponent},

  {path:'contact',component:ContactComponent},

  {path:'product',component:ProductComponent},

  {path:'',redirectTo:'home',pathMatch:'full'},

  {path:'\*\*',component:ErrorComponent}

];

@NgModule({

  imports: [RouterModule.forRoot(routes)],

  exports: [RouterModule]

})

export class AppRoutingModule { }

app.component.html

<div class="container">

<nav class="navbar navbar-default">

  <div class="container-fluid">

    <div class="navbar-header">

      <a class="navbar-brand" [routerLink]="['/']"><strong> {{title}} </strong></a>

    </div>

    <ul class="nav navbar-nav">

        <li><a [routerLink]="['home']">Home</a></li>

        <li><a [routerLink]="['product']">Product</a></li>

        <li><a [routerLink]="['contact']">Contact us</a></li>

    </ul>

  </div>

</nav>

 <router-outlet></router-outlet>

</div>

Angular Route Parameters: Passing Parameters to Route

**What are Route Parameters**

The Route parameters are a dynamic part of the Route and essential in determining the route.

For example, consider the following route

|  |  |
| --- | --- |
| 1  2  3 | { path: 'product', component: ProductComponent } |

The above route match only if the URL is /product

To retrieve the product details, our URL should look something like

/product/1  
/product/2

Where the second URL segment ( 1 and 2 ) being the id of the product. The **id** is dynamic and changes as per the selected Product. To handle such scenario[angular router](https://www.tektutorialshub.com/angular/angular-routing-navigation/) allows us to include **route parameters**, where we can send any dynamic value for a URL segment

**How to Pass parameters to Angular Route**

**Defining the Route**

We can define parameter by adding forward slash followed colon and a placeholder (id) as shown below

|  |  |
| --- | --- |
| 1  2  3 | { path: 'product/:id', component: ProductDetailComponent } |

Now above path matches the URLs /product/1 , /product/2, etc.

If you have more than one parameter, then you can extend it by adding one more forward slash followed colon and a placeholder

|  |  |
| --- | --- |
| 1  2  3 | { path: 'product/:id/:id1/:id2', component: ProductDetailComponent } |

The name id, id1 & id2 are placeholders for parameters. We will use them while retrieving the values of the parameters.

**app-routing.module.ts**

import { NgModule } from '@angular/core';

import { RouterModule, Routes } from '@angular/router';

import { AboutComponent } from './about/about.component';

import { ContactComponent } from './contact/contact.component';

import { ErrorComponent } from './error/error.component';

import { HomeComponent } from './home/home.component';

import { ProductDetailComponent } from './product-detail/product-detail.component';

import { ProductComponent } from './product/product.component';

const routes: Routes = [

  {path:'home',component:HomeComponent},

  {path:'about', component:AboutComponent},

  {path:'contact',component:ContactComponent},

  {path:'product',component:ProductComponent},

  {path:'product-detail/:id',component:ProductDetailComponent},

  {path:'',redirectTo:'home',pathMatch:'full'},

   {path:'\*\*',component:ErrorComponent}

];

@NgModule({

  imports: [RouterModule.forRoot(routes)],

  exports: [RouterModule]

})

export class AppRoutingModule { }

Product.component.html

<h1>Product List</h1>

<div class='table-responsive'>

    <table class='table'>

        <thead>

            <tr>

                <th>ID</th>

                <th>Name</th>

                <th>Price</th>

            </tr>

        </thead>

        <tbody>

            <tr \*ngFor="let product of products;">

                <td>{{product.productId}}</td>

                <td>

                    <a [routerLink]="['/product-detail',product.productId]">{{product.name}}</a>

                 </td>

                <td>{{product.price}}</td>

            </tr>

        </tbody>

    </table>

</div>

**product.service.ts**

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

import { Product } from './product';

@Injectable({

  providedIn: 'root'

})

export class ProductService {

  public getProducts(){

    let products:Product[];

    products=[

      new Product(1,'mac book',150000),

      new Product(2,'ipad air',60000),

      new Product(3,'iphone',139000)

    ]

    return products;

  }

  public getProduct(id:number)

  {

    let products:Product[]=this.getProducts();

    return products.find(p => p.productId==id);

}

  constructor() { }

}

**Product Details Component**

import { Component, OnInit,OnDestroy } from '@angular/core';

import { Product } from '../product';

import { ProductService } from '../product.service';

import { ProductComponent } from '../product/product.component';

import {Router,ActivatedRoute} from '@angular/router';

@Component({

  selector: 'app-product-detail',

  templateUrl: './product-detail.component.html',

  styleUrls: ['./product-detail.component.css']

})

export class ProductDetailComponent implements OnInit {

product:any='';

id:any='';

sub:any='';

constructor(private \_ActivatedRoute:ActivatedRoute,private \_router:Router,

     private \_productservice:ProductService) { }

  ngOnInit(): void {

     //approach :1

    this.id= this.\_ActivatedRoute.snapshot.params['id'];

    this.id= this.\_ActivatedRoute.snapshot.paramMap.get("id");

   this.product= this.\_productservice.getProduct(this.id)

   //approach:1

this.id =this.\_activatedRouted.snapshot.params['id'];

console.log('id '+this.id);

this.product= this.\_productservice.getProduct(this.id);

//approach :2

this.sub=this.\_activatedRouted.params.subscribe(params =>{

  this.id=params['id'];

  console.log(this.id);

  this.product=this.\_productservice.getProduct(this.id);

});

  }

//  ngOnDestroy()

//  {

//    this.sub.unsubscribe();

//  }

onBack()

{

  this.\_router.navigate(['product']);

}

}

**ParamMap**

The Angular adds the map all the route parameters in the ParamMap object, which can be accessed from the ActivatedRoute service

The [ParamMap](https://angular.io/api/router/ParamMap) makes it easier to work with parameters. We can use get or getAll methods to retrieve the value of the parameters in the component. Use the has method to check if a certain parameter exists.

The Older version of ActivatedRoute class has a Params array which is an array of the parameter values, indexed by name. You can still use it but It is now deprecated and is replaced by the ParamMap.

There are two ways in which you can use the ActivatedRoute to get the parameter value from the ParamMap object.

1. Using Snapshot
2. Using observable

**Using Snapshot**

|  |  |
| --- | --- |
| 1  2  3 | this.id=this.\_Activatedroute.snapshot.paramMap.get("id"); |

The snapshot property returns the initial value of the route. You can then access the paramsMap array, to access the value of the id, as shown above,

Use the Snapshot option, if you only need the initial value.

**Using Observable**

|  |  |
| --- | --- |
| 1  2  3  4  5 | this.\_Activatedroute.paramMap.subscribe(params => {      this.id = params.get('id');  }); |

You can retrieve the value of id by subscribing to the paramMap observable property of the activateRoute as shown above

Use this option if you expect the value of the parameter to change over time.

**Why use observable**

We usually retrieve the value of the parameter in the **[ngOninit](https://www.tektutorialshub.com/angular/angular-ngoninit-and-ngondestroy/)** life cycle hook, when the component initialised.

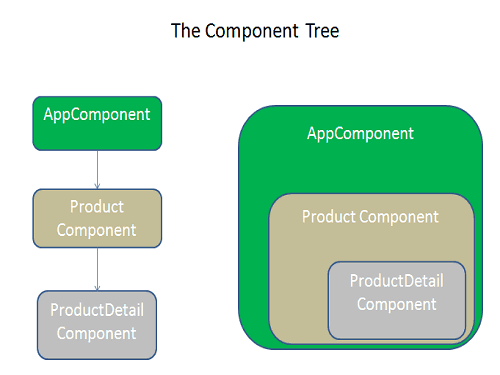
When the user navigates to the component again, the Angular does not create the new component but reuses the existing instance. In such circumstances, the ngOnInit method of the component is not called again. Hence you need a way to get the value of the parameter.

By subscribing to the observable paramMap property, you will retrieve the latest value of the parameter and update the component accordingly.

## Child Routes / Nested Routes

Example

Consider the following Component Tree



In the above example, the ProductComponent displays the list of Products. The ProductDetailsComponent is defined as the child of the ProductComponent displays the details of the selected Product.

App.routing .module.ts

import { NgModule } from '@angular/core';

import { RouterModule, Routes } from '@angular/router';

import { AboutComponent } from './about/about.component';

import { ContactComponent } from './contact/contact.component';

import { ErrorComponent } from './error/error.component';

import { HomeComponent } from './home/home.component';

import { ProductDetailComponent } from './product-detail/product-detail.component';

import { ProductComponent } from './product/product.component';

const routes: Routes = [

  {path:'home',component:HomeComponent},

  {path:'about', component:AboutComponent},

  {path:'contact',component:ContactComponent},

  {

    path:'product',component:ProductComponent,

children:[

  {path:'-detail/:id', component:ProductDetailComponent}

         ]

  }

,

  {path:'product-detail/:id',component:ProductDetailComponent},

  {path:'',redirectTo:'home',pathMatch:'full'},

   {path:'\*\*',component:ErrorComponent}

];

@NgModule({

  imports: [RouterModule.forRoot(routes)],

  exports: [RouterModule]

})

export class AppRoutingModule { }

product.component.html

<h1>Product List</h1>

<div class='table-responsive'>

    <table class='table'>

        <thead>

            <tr>

                <th>ID</th>

                <th>Name</th>

                <th>Price</th>

            </tr>

        </thead>

        <tbody>

            <tr \*ngFor="let product of products;">

                <td>{{product.productId}}</td>

                <td>

                    <a [routerLink]="['-detail',product.productId]">{{product.name}}</a>

                 </td>

                <td>{{product.price}}</td>

            </tr>

        </tbody>

    </table>

</div>

<router-outlet></router-outlet>

ng g c productoverview

ng g c productspec

app.routing.module.ts

import { NgModule } from '@angular/core';

import { RouterModule, Routes } from '@angular/router';

import { AboutComponent } from './about/about.component';

import { ContactComponent } from './contact/contact.component';

import { ErrorComponent } from './error/error.component';

import { HomeComponent } from './home/home.component';

import { ProductDetailComponent } from './product-detail/product-detail.component';

import { ProductComponent } from './product/product.component';

import { ProductoverviewComponent } from './productoverview/productoverview.component';

import { ProductspecComponent } from './productspec/productspec.component';

const routes: Routes = [

  {path:'home',component:HomeComponent},

  {path:'about', component:AboutComponent},

  {path:'contact',component:ContactComponent},

  {

    path:'product',component:ProductComponent,

children:[

  { path:'-detail/:id', component:ProductDetailComponent},

  {path:'overview',component:ProductoverviewComponent},

  {path:'spec',component:ProductspecComponent  }

         ]

  }

,

  //{path:'product-detail/:id',component:ProductDetailComponent},

  {path:'',redirectTo:'home',pathMatch:'full'},

   {path:'\*\*',component:ErrorComponent}

];

@NgModule({

  imports: [RouterModule.forRoot(routes)],

  exports: [RouterModule]

})

export class AppRoutingModule { }