topic

34. Angular Router Tutorial
35. Component Routing in Angular
36. Router Outlet in Angular
37. Multiple Router Outlets in Angular
38. Routing Strategy in Angular
39. Base Href in Angular
40. Router Module in Angular
41. Configure Component Routes in Angular
42. Parametrized Routes in Angular
43. RouterLink in Angular
44. Redirect Routes in Angular
45. Query Params in Angular
46. Wildcard Routes Link in Angular
47. Lazy Loading in Angular

Angular Router Tutorial

Angular Router by ARC

Angular router

- Routing is a mechanism used by Angular framework to manage the "paths" and "routes" of our Angular applications
- Routing strategy helps in navigation between various views in our Angular application.
- Angular framework comes with "Router" Module which has everything we need to design, develop and implement routes and navigation links
- Router is a singleton — which means there is ONLY one instance of the router in our Angular application

Routing in Angular

- Angular Router is the official Router module which is written and maintained by core Angular team
- The Router module is found in the package @angular/router
- We need to setup Router array — every time a request is made, the router will search in the list of array and find the most relevant match.
- Router has states- which helps us get important information about the current state and data related to routes
- We can handle various types of routes in Angular app
    - Routes for components
    - Getting Query Params from routes

    - Getting the URL segments
    - Loading child routes for a module
    - Lazy Loading
    - Handling wild card routes
    - Handling default routes
    - Handling 404 route

- All batteries included for Router

example of routing

- Home -> `http://myapplication.com/` -> Default Route
- Profile -> `http://myapplication.com/profile` -> Component Routing
- Search -> `http://myapplication.com/search?user=abc` -> Query Params
- Tasks -> `http://myapplication.com/tasks/10/category/pending` -> URL Segments
- Users -> `http://myapplication.com/users` -> Module
    - view-user -> `http://myapplication.com/users/view/10` -> Child Routes
    - edit-user -> `http://myapplication.com/users/edit/10` -> Child Routes
    - add-user -> `http://myapplication.com/users/add` -> Child Routes
    - manage-user -> `http://myapplication.com/users/manage` -> Child Routes
- PageNotFound -> `http://myapplication.com/pageNotFound` -> 404 error -> No matching routes

create AppRoutingModule in `app` folder

app.routing.module.ts

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

const routes: Routes = [
    // we will create all routes here
];

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

import AppRoutingModule in `app.module.ts`

app.module.ts

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { AppRoutingModule } from './app-routing.module';  //  importing routing
module
import { AppComponent } from './app.component';
import { HighlightDirective } from './highlight.directive';
import { UsersComponent } from './users/users.component';
import { FormsModule } from '@angular/forms' ;
@NgModule({
  declarations: [
    AppComponent,
    HighlightDirective,
    UsersComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModule, // routing module add in  imports array
    FormsModule,
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

Angular Router by Angular.io

Component Routing in Angular

Component Routing by ARC

Routes for components

- Each component can have its own Routes
- Various examples of **component routes** are:
    - /products
    - /products/view
    - /products/add
    - /users

create component `loans` , `loanTypes` , `add-loans`

```
D:\theory\Angular\ANGULAR BY ARC\project\simpleCRM>ng g c loans
CREATE src/app/loans/loans.component.html (20 bytes)
CREATE src/app/loans/loans.component.spec.ts (592 bytes)
CREATE src/app/loans/loans.component.ts (272 bytes)
CREATE src/app/loans/loans.component.scss (0 bytes)
UPDATE src/app/app.module.ts (817 bytes)
```

```
D:\theory\Angular\ANGULAR BY ARC\project\simpleCRM>ng g c loanTypes
CREATE src/app/loan-types/loan-types.component.html (25 bytes)
CREATE src/app/loan-types/loan-types.component.spec.ts (621 bytes)
CREATE src/app/loan-types/loan-types.component.ts (291 bytes)
CREATE src/app/loan-types/loan-types.component.scss (0 bytes)
UPDATE src/app/app.module.ts (913 bytes)

D:\theory\Angular\ANGULAR BY ARC\project\simpleCRM>ng g c add-loans
CREATE src/app/add-loans/add-loans.component.html (24 bytes)
CREATE src/app/add-loans/add-loans.component.spec.ts (614 bytes)
CREATE src/app/add-loans/add-loans.component.ts (287 bytes)
CREATE src/app/add-loans/add-loans.component.scss (0 bytes)
UPDATE src/app/app.module.ts (1005 bytes)
```

app.routing.module.ts

```
import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';
import { AddLoansComponent } from './add-loans/add-loans.component';
import { LoanTypesComponent } from './loan-types/loan-types.component';
import { LoansComponent } from './loans/loans.component';

const routes: Routes = [
  // we will create all routes here
  {
    path: 'loans',
    component: LoansComponent
  },
  {
    path: 'loans/add-loan',
    component: AddLoansComponent
  },
  {
    path: 'loans-types',
    component: LoanTypesComponent
  }
];

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

app.component.html

```
<h1 class="c1">{{title}}</h1>

<h4> Angular Chaining Pipes </h4>
```

```
<div> date Pipe :- {{ dateExampIe | date | uppercase }}</div>
<div> currencyExample Pipe :- CANADIAN  DOLLARS :- {{ currencyExample | currency :
'CAD' | lowercase}} </div>

<router-outlet></router-outlet> <!-- HERE EVERY COMPONENT WILL SHOW ON ROUTER -->
<!-- here all routing component will showin view -->
```

app.component.ts

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

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  // template: `<h1>{{title}}</h1>`,
  styleUrls: ['./app.component.scss']
})

export class AppComponent {

  title = 'simpleCRM';

  userName = ""

  lowerCaseExamp1e = "ARC TUTORIALS";
  upperCaseExamp1e = "learn ANgular framework tutorials";
  dateExampIe = Date.now();
  jsonExamp1e = {username: "arc" , major: "Angular" , experience : "lem"}
  currencyExample = 125
  percentExample = 0.6767
}
```

loans.component.html

```
<p>loans works!</p>
```

loans.component.ts

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

@Component({
  selector: 'app-loans',
  templateUrl: './loans.component.html',
  styleUrls: ['./loans.component.scss']
})
export class LoansComponent implements OnInit {
```

```
  constructor() { }

  ngOnInit(): void {
  }

}
```

add-loans.component.html

```
<p>add-loans works!</p>
```

add-loans.component.ts

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

@Component({
  selector: 'app-add-loans',
  templateUrl: './add-loans.component.html',
  styleUrls: ['./add-loans.component.scss']
})
export class AddLoansComponent implements OnInit {

  constructor() { }

  ngOnInit(): void {
  }

}
```

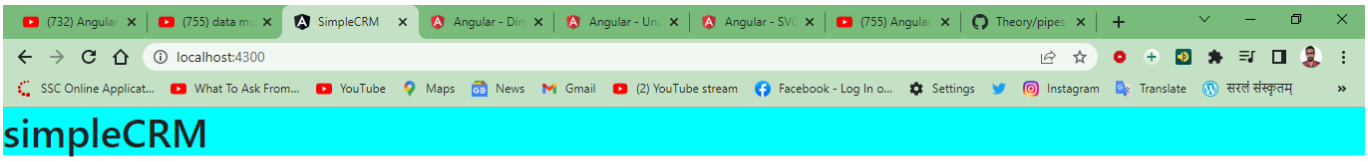
loan-types.component.html

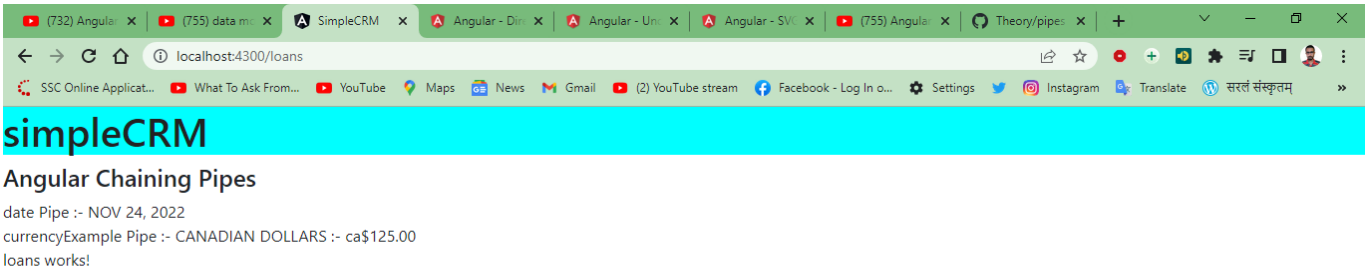
```
<p>loan-types works!</p>
```

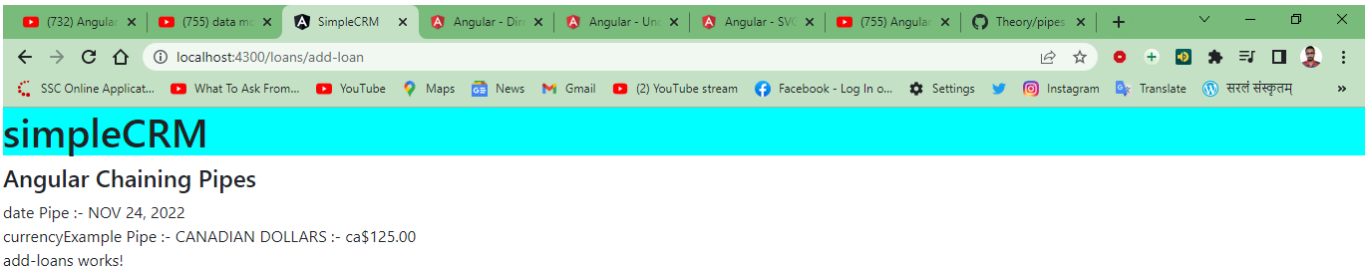
loan-types.component.ts

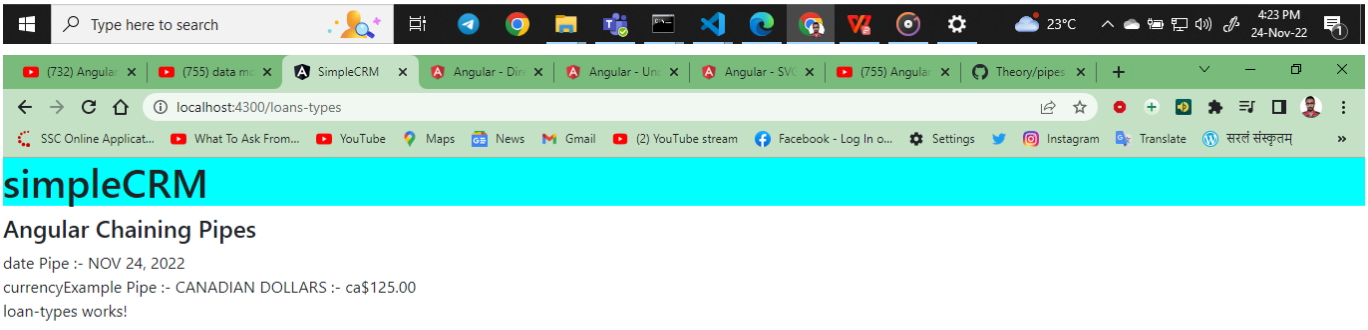
```
import { Component, OnInit } from '@angular/core';

@Component({
  selector: 'app-loan-types',
  templateUrl: './loan-types.component.html',
  styleUrls: ['./loan-types.component.scss']
})
export class LoanTypesComponent implements OnInit {
```

```
    constructor() { }

    ngOnInit(): void {
    }

  }
```









Commmon Mistakes

- developers will add "/" in path
- component name in qoutes
- bad formed array of routes
- Sometimes your editor is not importing component correctly
- Red color underline means there is some error

Component Routing by Angular.io

Router Outlet in Angular

Router Outlet by ARC

introduction :-

- The Router-Outlet is a directive
  - that's available from the router library
  - where the Router inserts the component
  - that gets matched based on the current browser's URL.
- You can add multiple outlets in your Angular application
  - which enables you to implement advanced routing scenarios.
- By default there is always one router outlet defined — in `app.component.html`

notes :-

1. Router outlet is a built-in directive
2. Every Angular app should have "atleast" 1 router outlet -> primary router outlet
3. By default - the router outlet is defined in app.component.html file
4. Router outlet will match the matching routes for the components
   - takes its output
   - inside inside the page
5. Multiple router outlets in application
   - We can have more than 1 router outlet

Common Mistakes

- you don't router-outlet `<router-outlet></router-outlet>`
  - you wont see the output
- Best practice is to leave router-outlet empty

EXAMPLE :-

app.routing.module.ts

```
import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';
import { AddLoansComponent } from './add-loans/add-loans.component';
import { LoanTypesComponent } from './loan-types/loan-types.component';
import { LoansComponent } from './loans/loans.component';

const routes: Routes = [
  // we will create all routes here
```

```
  {
    path: 'loans',
    component: LoansComponent
  },
  {
    path: 'loans/add-loan',
    component: AddLoansComponent
  },
  {
    path: 'loans-types',
    component: LoanTypesComponent
  }
];

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

app.component.html

```
<h1 class="c1">{{title}}</h1>

<h4> Angular Chaining Pipes </h4>

<div> date Pipe :- {{ dateExampIe | date | uppercase }}</div>
<div> currencyExample Pipe :- CANADIAN  DOLLARS :- {{ currencyExample | currency :
'CAD' | lowercase}} </div>

<router-outlet></router-outlet> <!-- HERE EVERY COMPONENT WILL SHOW ON ROUTER -->
<!-- here all routing component will showin view -->
```

app.component.ts

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

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  // template: `<h1>{{title}}</h1>`,
  styleUrls: ['./app.component.scss']
})

export class AppComponent {

  title = 'simpleCRM';

  userName = ""
```

```
  lowerCaseExamp1e = "ARC TUTORIALS";
  upperCaseExamp1e = "learn ANgular framework tutorials";
  dateExampIe = Date.now();
  jsonExamp1e = {username: "arc" , major: "Angular" , experience : "lem"}
  currencyExample = 125
  percentExample = 0.6767
}
```

loans.component.html

```
<p>loans works!</p>
```

loans.component.ts

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

@Component({
  selector: 'app-loans',
  templateUrl: './loans.component.html',
  styleUrls: ['./loans.component.scss']
})
export class LoansComponent implements OnInit {

  constructor() { }

  ngOnInit(): void {
  }

}
```

add-loans.component.html

```
<p>add-loans works!</p>
```

add-loans.component.ts

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

@Component({
  selector: 'app-add-loans',
  templateUrl: './add-loans.component.html',
  styleUrls: ['./add-loans.component.scss']
})
export class AddLoansComponent implements OnInit {
```

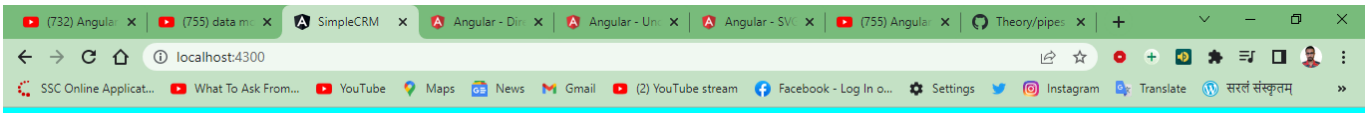
```
  constructor() { }

  ngOnInit(): void {
  }

}
```

loan-types.component.html

```
<p>loan-types works!</p>
```

loan-types.component.ts

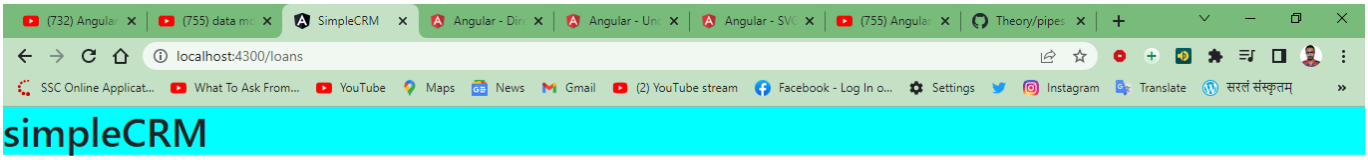
```
import { Component, OnInit } from '@angular/core';

@Component({
  selector: 'app-loan-types',
  templateUrl: './loan-types.component.html',
  styleUrls: ['./loan-types.component.scss']
})
export class LoanTypesComponent implements OnInit {

  constructor() { }

  ngOnInit(): void {
  }

}
```



simpleCRM

Angular Chaining Pipes

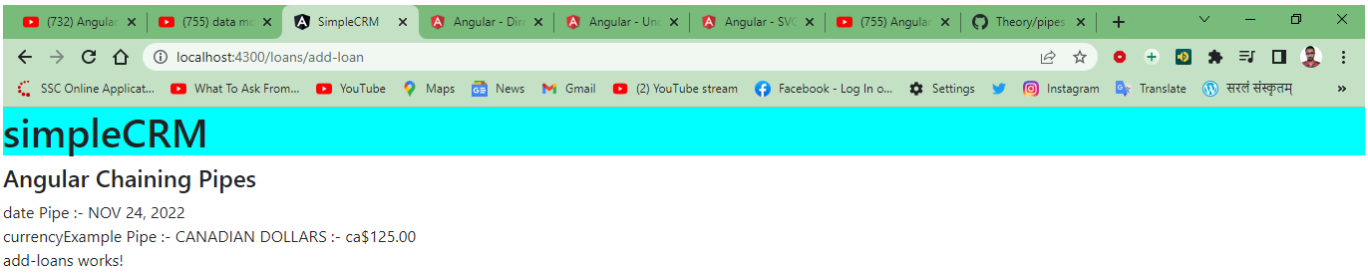
date Pipe :- NOV 24, 2022
currencyExample Pipe :- CANADIAN DOLLARS :- ca$125.00

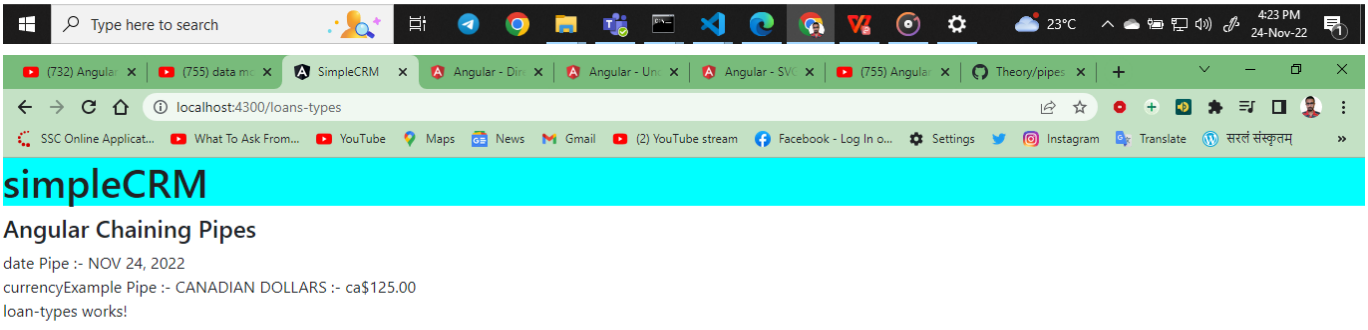


simpleCRM

Angular Chaining Pipes

date Pipe :- NOV 24, 2022
currencyExample Pipe :- CANADIAN DOLLARS :- ca$125.00
loans works!





Router Outlet by Angular.io

Multiple Router Outlets in Angular

Multiple Router Outlets by ARC

NOTE :-

- Multiple Router Outlets

- The Router-Outlet is a directive that's available from the router library where the Router inserts tutorials component that gets matched based on the current browser's URL.
- You can add multiple outlets in your Angular application which enables you to implement advanced routing scenarios.
- By default — there is always a router-outlet and it's treated as "primary"
- We need to define named router outlets
- **Example of declaring multiple router outlets**

```
{
    path: 'add',
    component: AddLoansComponent,
    outlet:'route1'
}
```

- **http://localhost:4200/loans(route1 :add)**

importanr points :-

1. We can have multiple router Outlets
2. by default there is `always/"atleast"` 1 router outlet in `app.component.html` file
3. When we don't provide any name for `router-outlet` , it becomes primary.
4. There should be only 1 primary
5. We can define multiple router outlets by giving name to them
6. That's why we call them "named" router outlets
7. we can give any name we want - give meaningful names
8. In routing module
    - if you don't define outlet - it means its primary
9. It will NOT show if you directly access it in the URL

---

1. Synatx should be like this
    - `http://localhost:4200/<primary-route>( <routerOutletName> : <secondaryPath> )`
2. Why are using this?
    - Avoid this use case in applications?
    - You can inject components
3. URL is not user friendly
    - bookmarable URL

4. I have not personally seen this used a lot
    - It's not used very much

EXAMPLE :-

app.routing.module.ts

```
import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';
import { AddLoansComponent } from './add-loans/add-loans.component';
import { LoanTypesComponent } from './loan-types/loan-types.component';
import { LoansComponent } from './loans/loans.component';

/*
const routes: Routes = [
  // we will create all routes here
  {
    path: 'loans',
    component: LoansComponent
  },
  {
    path: 'loans/add-loan',
    component: AddLoansComponent
  },
  {
    path: 'loans-types',
    component: LoanTypesComponent
  }
];
 */


const routes: Routes = [
  // we will create multiple routes here
  {
    path: 'loans',
    component: LoansComponent
  },
  {
    path: 'add',
    component: AddLoansComponent,
    outlet:'addLoan'
  },
  {
    path: 'types',
    component: LoanTypesComponent,
    outlet:'editLoan'
  }
];

@NgModule({
  imports: [RouterModule.forRoot(routes)],
```

```
  exports: [RouterModule]
})
export class AppRoutingModule { }
```

app.component.html

```
<h1 class="c1">{{title}}</h1>

<h4> Angular Chaining Pipes </h4>

<div> date Pipe :- {{ dateExampIe | date | uppercase }}</div>
<div> currencyExample Pipe :- CANADIAN  DOLLARS :- {{ currencyExample | currency :
'CAD' | lowercase}} </div>

<!-- HERE EVERY COMPONENT WILL SHOW ON ROUTER -->
<router-outlet></router-outlet>  <!-- primary outlet -->
<router-outlet name="addLoan"></router-outlet>   <!-- named router outlet -->
<router-outlet name="editLoan"></router-outlet>   <!-- named router outlet -->
```

app.component.ts

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

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  // template: `<h1>{{title}}</h1>`,
  styleUrls: ['./app.component.scss']
})

export class AppComponent {

  title = 'simpleCRM';

  userName = ""

  lowerCaseExamp1e = "ARC TUTORIALS";
  upperCaseExamp1e = "learn ANgular framework tutorials";
  dateExampIe = Date.now();
  jsonExamp1e = {username: "arc" , major: "Angular" , experience : "lem"}
  currencyExample = 125
  percentExample = 0.6767
}
```

loans.component.html

```
<p>loans works!</p>
```

loans.component.ts

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

@Component({
  selector: 'app-loans',
  templateUrl: './loans.component.html',
  styleUrls: ['./loans.component.scss']
})
export class LoansComponent implements OnInit {

  constructor() { }

  ngOnInit(): void {
  }

}
```

add-loans.component.html

```
<p>add-loans works!</p>
```

add-loans.component.ts

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

@Component({
  selector: 'app-add-loans',
  templateUrl: './add-loans.component.html',
  styleUrls: ['./add-loans.component.scss']
})
export class AddLoansComponent implements OnInit {
  constructor() { }
  ngOnInit(): void {
  }
}
```

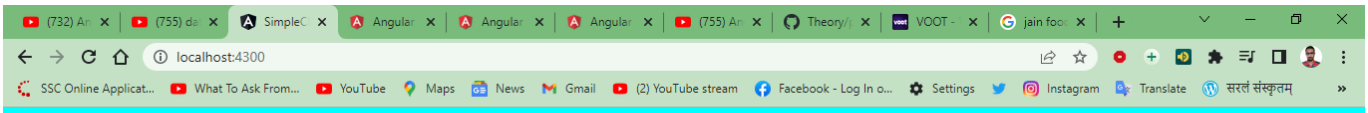
loan-types.component.html

```
<p>loan-types works!</p>
```

loan-types.component.ts

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

@Component({
  selector: 'app-loan-types',
  templateUrl: './loan-types.component.html',
  styleUrls: ['./loan-types.component.scss']
})
export class LoanTypesComponent implements OnInit {

  constructor() { }

  ngOnInit(): void {
  }

}
```

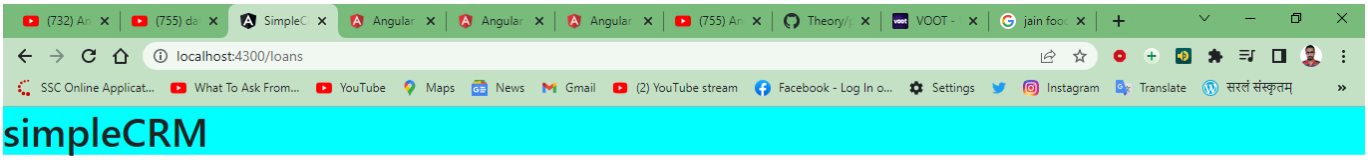
- **Syntax** :- `http://localhost:4200/<primary-route>( <routerOutletName> : <secondaryPath> )`
- Example:-
    - http://localhost:4300
    - http://localhost:4300/loans
    - http://localhost:4300/loans(editLoan:types)
    - http://localhost:4300/loans(addLoan:add)



simpleCRM

Angular Chaining Pipes

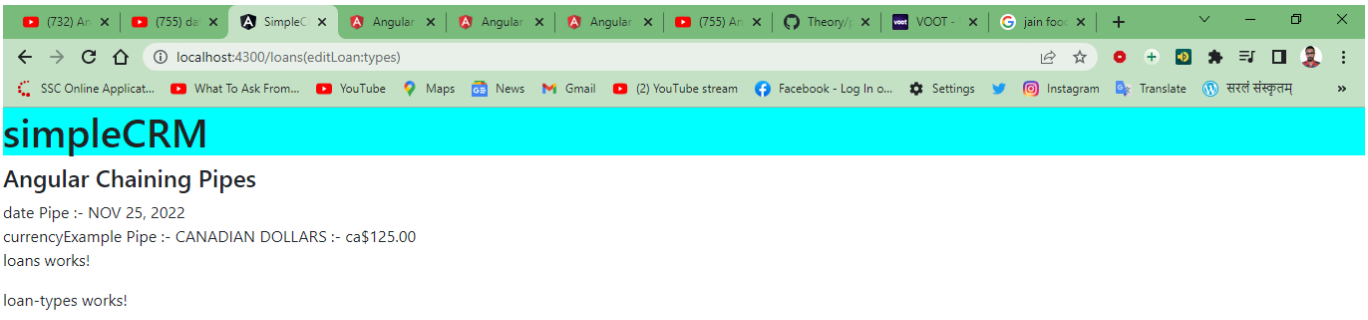
date Pipe :- NOV 25, 2022
currencyExample Pipe :- CANADIAN DOLLARS :- ca$125.00

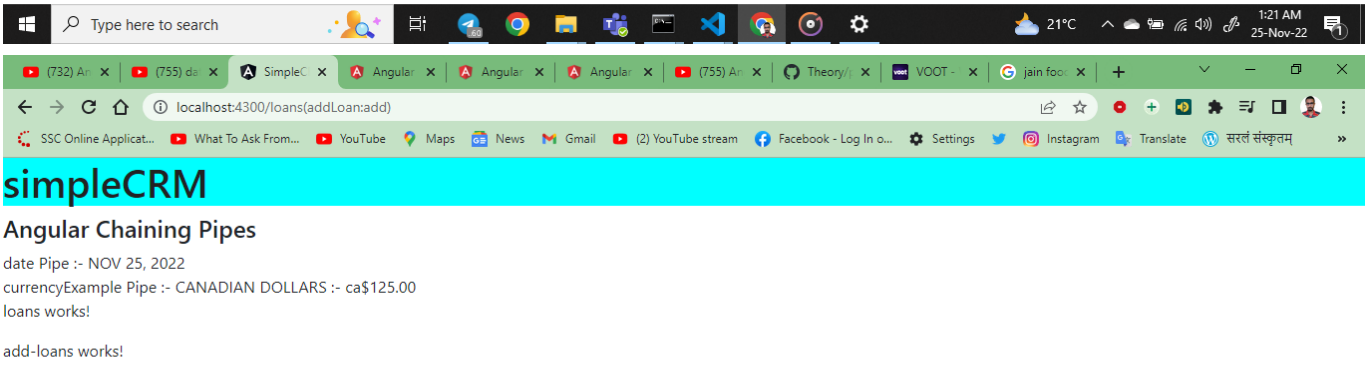


simpleCRM

Angular Chaining Pipes

date Pipe :- NOV 25, 2022
currencyExample Pipe :- CANADIAN DOLLARS :- ca$125.00
loans works!





Multiple Router Outlets by Angular.io

Routing Strategy in Angular

Routing Strategy by ARC

note :-

- Before we start implementing our routes in our application, its important to understand and plan what will be our routing strategy

    - `import { LocationStrategy } from '@angular/common';`

- We need to add this in Providers of our Module
    - `{provide: LocationStrategy, useCIass: HashLocationStrategy}`
- Angular provides 2 types of routing strategy we can use:
    - `PathLocationStrategy`
    - `HashLocationStrategy`
- **By default** —Angular makes use of the `PathLocationStrategy`
- with `PathLocationStrategy` - wé will see the # in the URL

IMPORTANTS :-

1. Routing behaviour of the applications URLs
2. Angular provies 2 routing strategies

- PathLocationStrategy
- HashLocationStrategy

PathLocationStrategy

- Default routing strategy for Angular apps
- HTML 5 push state URL
- Examples
    - http://myapp.com/dashboard
    - http://myapp.com/user/10
    - http://myapp.com/user/10/photos
    - http://myapp.com/search?query=abc&state=ka&city=bengaluru

HashLocationStrategy

- URL segments/patterns
- URLs will have hash in the URLs
- Examples
    - http://myapp.com/#/dashboard
    - http://myapp.com/#/user/10
    - http://myapp.com/#/user/10/photos
    - http://myapp.com/#/search?query=abc&state=ka&city=bengaluru

1. Hands-on examples for PathLocationStrategy
    - Default behaviour of Angular apps
2. Hands-on examples for HashLocationStrategy
    - We need to import HashLocationStrategy from @angular/core
    - Add it to Providers array

    - Angular will start loading our URLs using `#`

3. Why do we need 2 different types of routing?
    - Angular is a SPA(single page app)
        - index.html
    - Cloud vendors
        - AWS
        - GCP
        - Azure
        - Hosting Provider (Bluehost, Siteground, DigitalOcean)
    - `/#/loans/add -> Route`
    - `index.html/#/loans/add`
4. Which one you should use when?
    - Really there is no difference affect your application
    - PathLocationStrategy
        - Clean URLs
        - Simple
        - Bookmarbale
        - Easy to Remember

Routing with `PathLocationStrategy`

- Default HTML 5 push state
- Various examples of PathLocationStrategy are given here:
    - `/products`
    - `/product/10`
    - `/product/10/details`
    - `/product?search=param1`

Routing with `HashLocationStrategy`

- Using the `#` URL segments
- Various examples of `HashLocationStrategy` are given here:
    - `#/products`
    - `#/product/10`
    - `#/product/l O/details`
    - `#/product?search=paraml`

EXAMPLE :-

app.modue.ts

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { AppRoutingModule } from './app-routing.module';  //  importing routing
module
import { AppComponent } from './app.component';
import { HighlightDirective } from './highlight.directive';
import { UsersComponent } from './users/users.component';
import { FormsModule } from '@angular/forms';
import { LoansComponent } from './loans/loans.component';
import { LoanTypesComponent } from './loan-types/loan-types.component';
import { AddLoansComponent } from './add-loans/add-loans.component'
import { HashLocationStrategy, LocationStrategy } from '@angular/common'; //
import routing strategy here

@NgModule({
  declarations: [
    AppComponent,
    HighlightDirective,
    UsersComponent,
    LoansComponent,
    LoanTypesComponent,
    AddLoansComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModule, // routing module add in  imports array
    FormsModule,
  ],
  providers: [
    { provide: LocationStrategy , useClass:HashLocationStrategy } , // add routing
strategy in providers array
    // {provide: LocationStrategy , useClass:PathLocationStrategy} // add routing
strategy in providers array
  ],
  bootstrap: [AppComponent]
})

export class AppModule { }
```

app.routing.module.ts

```
import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';
import { AddLoansComponent } from './add-loans/add-loans.component';
import { LoanTypesComponent } from './loan-types/loan-types.component';
import { LoansComponent } from './loans/loans.component';

/*
const routes: Routes = [
  // we will create all routes here
  {
```

```
    path: 'loans',
    component: LoansComponent
  },
  {
    path: 'loans/add-loan',
    component: AddLoansComponent
  },
  {
    path: 'loans-types',
    component: LoanTypesComponent
  }
];
 */


const routes: Routes = [
  // we will create multiple routes here
  {
    path: 'loans',
    component: LoansComponent
  },
  {
    path: 'add',
    component: AddLoansComponent,
    outlet:'addLoan'
  },
  {
    path: 'types',
    component: LoanTypesComponent,
    outlet:'editLoan'
  }
];

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

app.component.html

```
<h1 class="c1">{{title}}</h1>

<h4> Angular Chaining Pipes </h4>

<div> date Pipe :- {{ dateExampIe | date | uppercase }}</div>
<div> currencyExample Pipe :- CANADIAN  DOLLARS :- {{ currencyExample | currency :
'CAD' | lowercase}} </div>

<!-- HERE EVERY COMPONENT WILL SHOW ON ROUTER -->
<router-outlet></router-outlet>  <!-- primary outlet -->
<router-outlet name="addLoan"></router-outlet>   <!-- named router outlet -->
```

```
<router-outlet name="editLoan"></router-outlet>   <!-- named router outlet -->
```

app.component.ts

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

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  // template: `<h1>{{title}}</h1>`,
  styleUrls: ['./app.component.scss']
})

export class AppComponent {

  title = 'simpleCRM';

  userName = ""

  lowerCaseExamp1e = "ARC TUTORIALS";
  upperCaseExamp1e = "learn ANgular framework tutorials";
  dateExampIe = Date.now();
  jsonExamp1e = {username: "arc" , major: "Angular" , experience : "lem"}
  currencyExample = 125
  percentExample = 0.6767
}
```

loans.component.html

```
<p>loans works!</p>
```

loans.component.ts

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

@Component({
  selector: 'app-loans',
  templateUrl: './loans.component.html',
  styleUrls: ['./loans.component.scss']
})
export class LoansComponent implements OnInit {

  constructor() { }

  ngOnInit(): void {
  }
```

```
}
```

add-loans.component.html

```
<p>add-loans works!</p>
```

add-loans.component.ts

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

@Component({
  selector: 'app-add-loans',
  templateUrl: './add-loans.component.html',
  styleUrls: ['./add-loans.component.scss']
})
export class AddLoansComponent implements OnInit {
  constructor() { }
  ngOnInit(): void {
  }
}
```

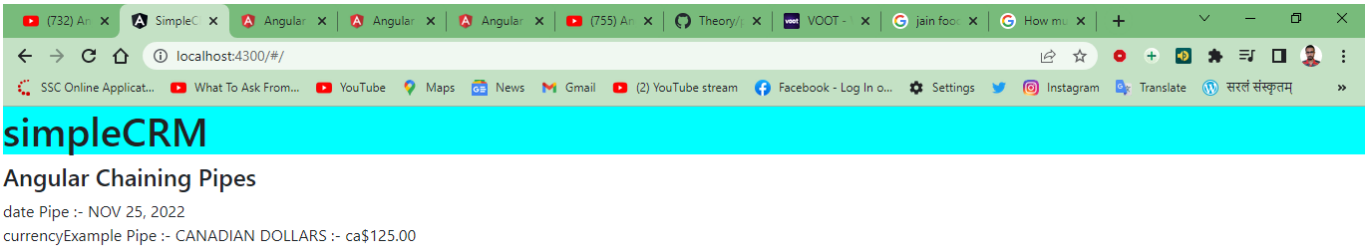
loan-types.component.html

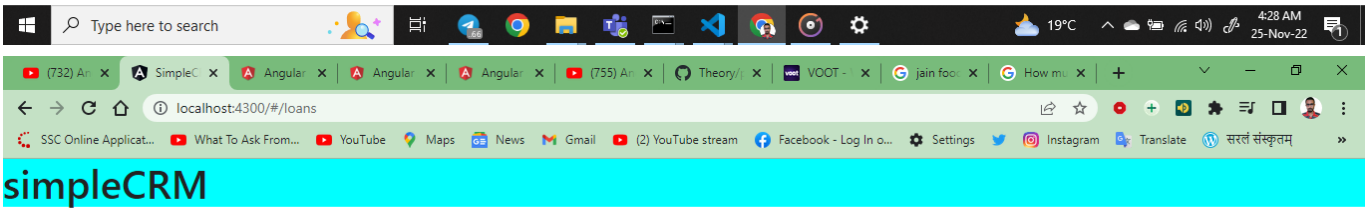
```
<p>loan-types works!</p>
```

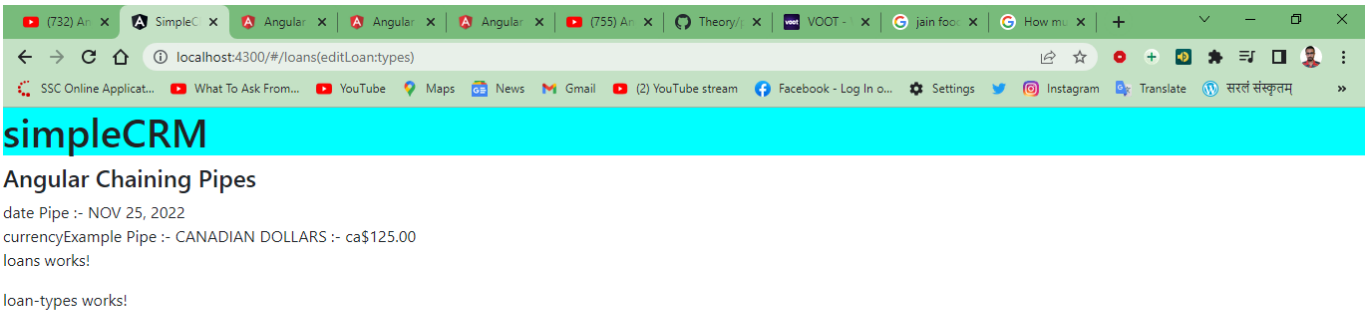
loan-types.component.ts

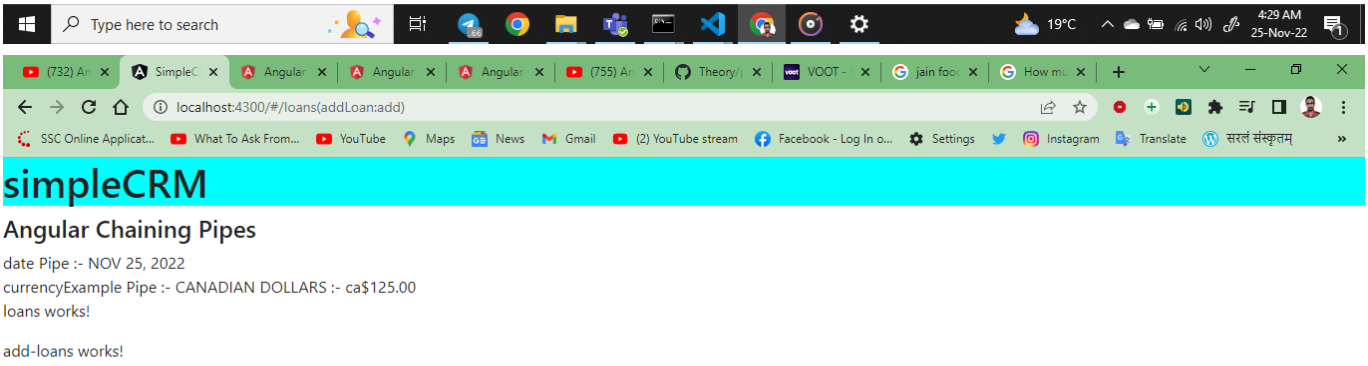
```
import { Component, OnInit } from '@angular/core';

@Component({
  selector: 'app-loan-types',
  templateUrl: './loan-types.component.html',
  styleUrls: ['./loan-types.component.scss']
})
export class LoanTypesComponent implements OnInit {

  constructor() { }

  ngOnInit(): void {
  }

}
```

- **Syntax** :- `http://localhost:4200/<primary-route>( <routerOutletName> : <secondaryPath> )`
- Example:-
    - http://localhost:4300/#/
    - http://localhost:4300/#/loans
    - http://localhost:4300/#/loans(editLoan:types)
    - http://localhost:4300/#/loans(addLoan:add)









Routing Strategy by Angular.io

Base Href in Angular

Base Href by ARC

Routing — Base Href

- Every Angular application has MANDATORY base href
- Angular application is a **SPA** (Single Page Architecture)

    - which means there will be only one HTML file
- The default base href is set to "`/`" the root folder
- The Base HREF is present in `index.html` file for all Angular applications

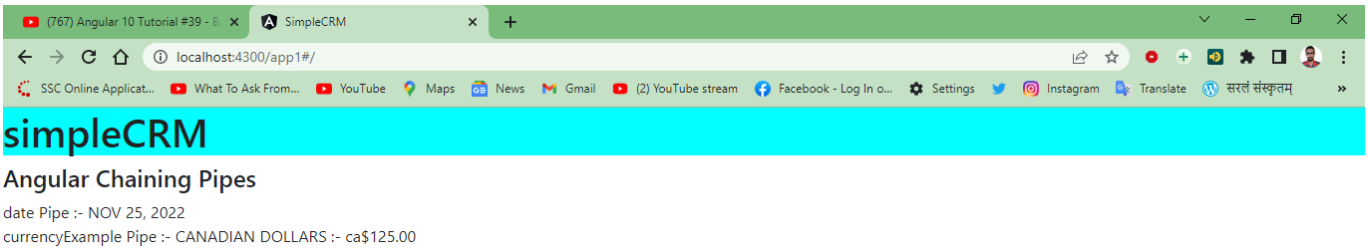
Routing — Base Href

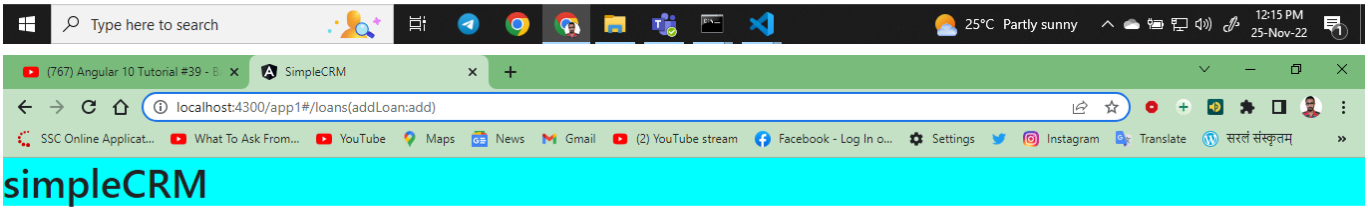
- Wrong configuration leads to pointing to wrong folder root path
- Setting the base href using the command line —`base-href=`
- **Syntax**:
    - `<base href="/" >` in **index.html**

example :-

index.html if `<base href="/app1">`

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>SimpleCRM</title>
  <!-- <base href="/"> -->
  <base href="/app1">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="icon" type="image/x-icon" href="favicon.ico">
</head>
<body>
  <app-root></app-root>
</body>
</html>
```

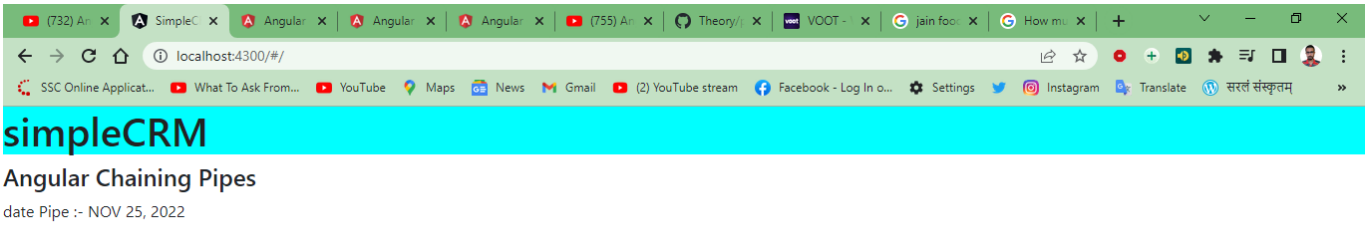




index.html if `<base href="/">`

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>SimpleCRM</title>
  <base href="/">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="icon" type="image/x-icon" href="favicon.ico">
</head>
```

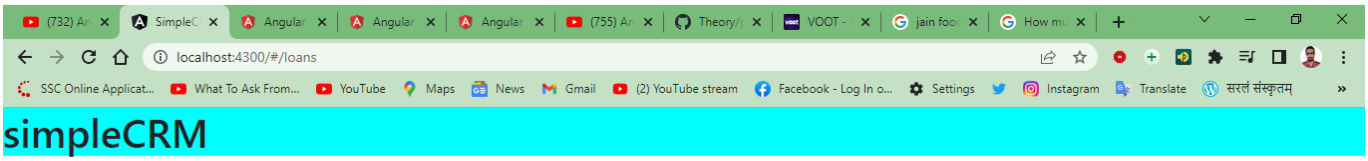
```
<body>
  <app-root></app-root>
</body>
</html>
```



simpleCRM

Angular Chaining Pipes

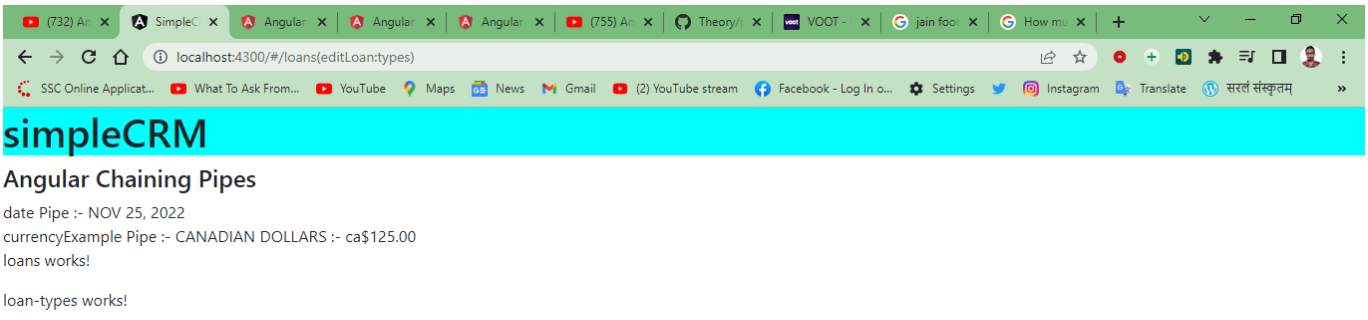
date Pipe :- NOV 25, 2022
currencyExample Pipe :- CANADIAN DOLLARS :- ca$125.00

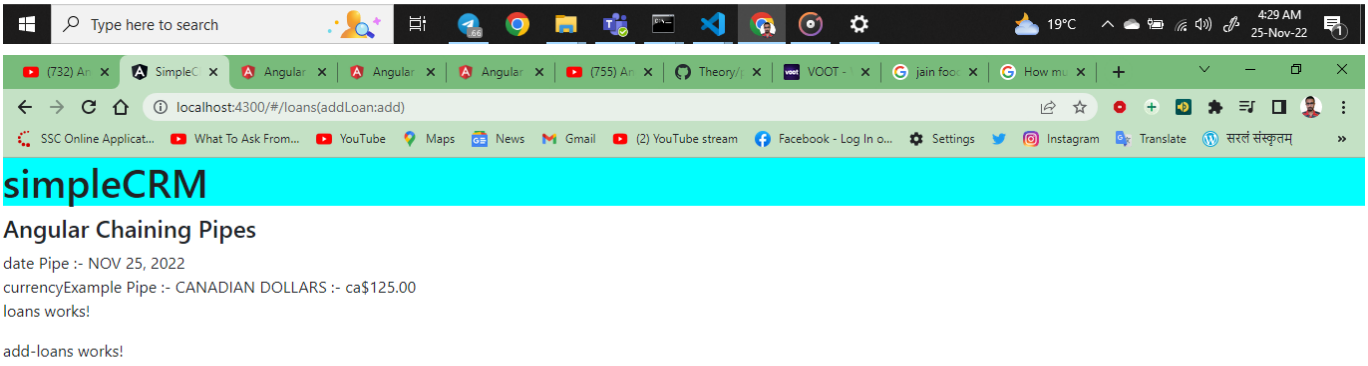


simpleCRM

Angular Chaining Pipes

date Pipe :- NOV 25, 2022
currencyExample Pipe :- CANADIAN DOLLARS :- ca$125.00
loans works!





Base Href by Angular.io

Router Module in Angular

Router Module by ARC

Routing Module

- Routing Module is a placeholder for configuring all routes and navigations in one module
- Best practice is to have all routes configured in one place

- Easy to maintain and debug
- We can generate the app routing module using the CLI
    - `ng generate module app-routing --flat –module=app`

introduction :-

1. Its a single module and placeholder where all our routes are configured for that particular module
2. Each module can have its own routes
3. During the angular app installation
    - we get an option - Do you want to have routing in yoru application?
        - it will automatically create the app-routing module file for us
4. `ng g module app-routing --flat --module=app`

```
D:\theory\Angular\ANGULAR BY ARC\project\simpleCRM> ng g module app-routing --flat
--module=app
CREATE src/app/app-routing.module.ts (196 bytes)
UPDATE src/app/app.module.ts (1173 bytes)
D:\theory\Angular\ANGULAR BY ARC\project\simpleCRM>
```

Router Module

- We need to import modules from the package

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

- We need to configure route path array in the file

```
const routes: Routes = [];
```

- Then we need to define our module

```
@NgModuIe({
imports: [RouterModule.forRoot(routes)],
exports: [RouterModule]
})
```

- We need to export the module

```
export class AppRoutingModule { }
```

- Import the module in the `AppModule` file

example :-

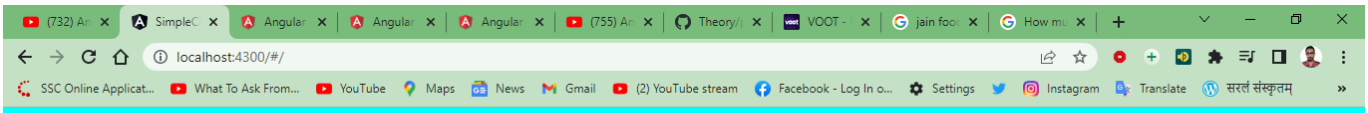
app-routing.module.ts

```
import { CommonModule } from '@angular/common';
import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';
import { AddLoansComponent } from './add-loans/add-loans.component';
import { LoanTypesComponent } from './loan-types/loan-types.component';
import { LoansComponent } from './loans/loans.component';

const routes: Routes = [
  // we will create all routes here
  {
    path: 'loans',
    component: LoansComponent
  },
  {
    path: 'add',
    component: AddLoansComponent,
    outlet:'addLoan'
  },
  {
    path: 'types',
    component: LoanTypesComponent,
    outlet:'editLoan'
  }
];



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

app.module.ts

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

import { AppRoutingModule } from './app-routing.module';  //  importing routing
module
```

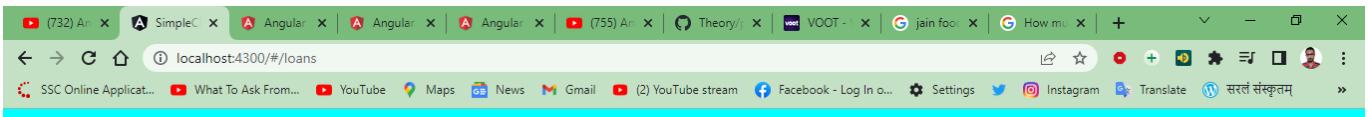
```
import { AppComponent } from './app.component';
import { HighlightDirective } from './highlight.directive';
import { UsersComponent } from './users/users.component';
import { FormsModule } from '@angular/forms';
import { LoansComponent } from './loans/loans.component';
import { LoanTypesComponent } from './loan-types/loan-types.component';
import { AddLoansComponent } from './add-loans/add-loans.component'
import { HashLocationStrategy, LocationStrategy , PathLocationStrategy } from
'@angular/common'; // import routing strategy here

@NgModule({
  declarations: [
    AppComponent,
    HighlightDirective,
    UsersComponent,
    LoansComponent,
    LoanTypesComponent,
    AddLoansComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModule, // routing module add in  imports array
    FormsModule,


  ],
  providers: [
    {provide: LocationStrategy , useClass:HashLocationStrategy} , // add routing
strategy in providers array
    // {provide: LocationStrategy , useClass:PathLocationStrategy} // add routing
strategy in providers array
  ],

  bootstrap: [AppComponent]
})
export class AppModule { }
```



simpleCRM

Angular Chaining Pipes

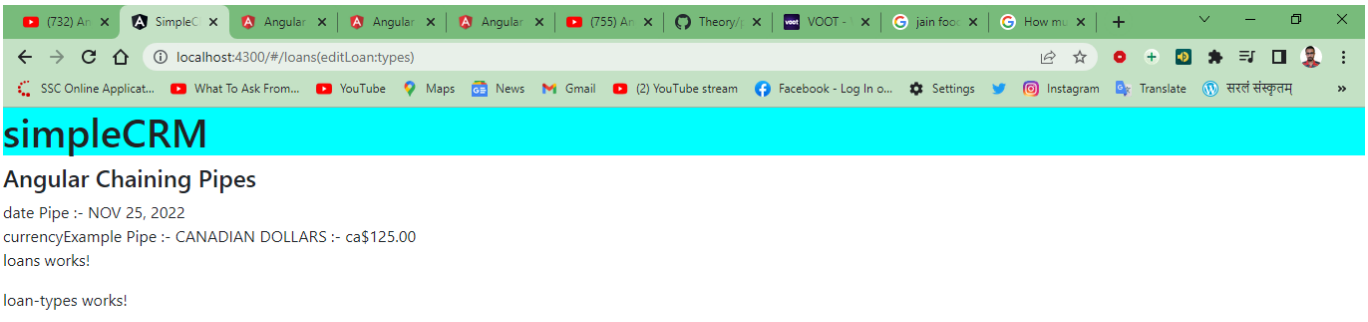
date Pipe :- NOV 25, 2022
currencyExample Pipe :- CANADIAN DOLLARS :- ca$125.00

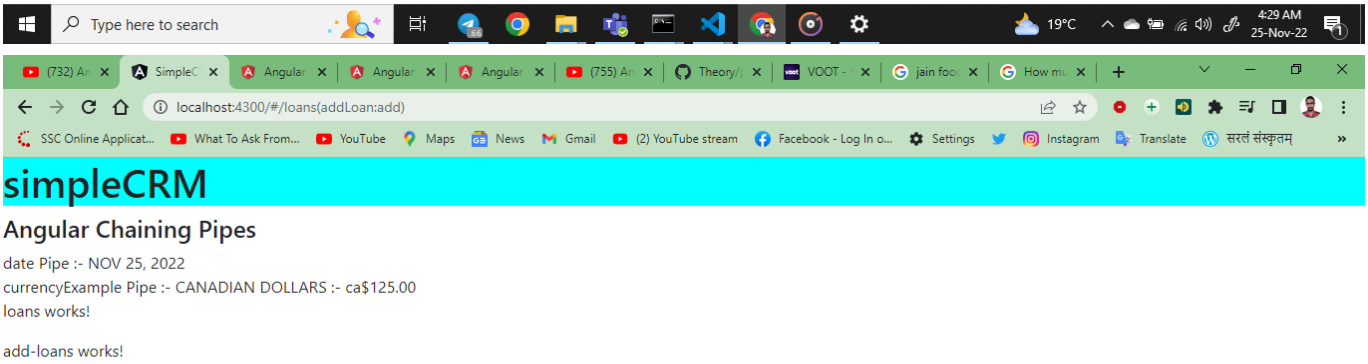


simpleCRM

Angular Chaining Pipes

date Pipe :- NOV 25, 2022
currencyExample Pipe :- CANADIAN DOLLARS :- ca$125.00
loans works!





Router Module by Angular.io

Configure Component Routes in Angular

Configure Component Routes by ARC

Component Routes - Configuring Routes

- We can configure routes to redirect route for various paths .
    - Path

    - Component
    - redirectTo
    - Children
    - Outlet
    - pathMatch
- Let's learn how to configure routes in the routing module .

notes :-

1. There are various options that we can configure in Component Routes
2. Some of the ones that we have seem in previous/earlier tutorials ar earlier are
    - path
    - component
    - Router Outlet Tutorial - Please check the playlist
        - outlet
    - children
    - redirectTo ->
    - pathMatch -> Will cover in coming episodes

Component Routes - Configuring Routes

- Create a Routes Array in App Routing module

```
 const routes : Routes = [
    { path : '' , redirectTo: 'home', pathMatch: 'full'},
    { path : 'home', component : componentName },
    { path : 'dashboard', component : componentName2 },
    { path : 'terms', component : componentName3 },
    { path : '**' , redirectTo: 'enroll', pathMatch: 'full'}
]
```

EXAMPLE :-

app.modue.ts

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { AppRoutingModule } from './app-routing.module';  //  importing routing
module
import { AppComponent } from './app.component';
import { HighlightDirective } from './highlight.directive';
import { UsersComponent } from './users/users.component';
import { FormsModule } from '@angular/forms';
import { LoansComponent } from './loans/loans.component';
import { LoanTypesComponent } from './loan-types/loan-types.component';
import { AddLoansComponent } from './add-loans/add-loans.component'
import { HashLocationStrategy, LocationStrategy } from '@angular/common'; //
```

```
import routing strategy here

@NgModule({
  declarations: [
    AppComponent,
    HighlightDirective,
    UsersComponent,
    LoansComponent,
    LoanTypesComponent,
    AddLoansComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModule, // routing module add in  imports array
    FormsModule,
  ],
  providers: [
    { provide: LocationStrategy , useClass:HashLocationStrategy } , // add routing
strategy in providers array
    // {provide: LocationStrategy , useClass:PathLocationStrategy} // add routing
strategy in providers array
  ],
  bootstrap: [AppComponent]
})

export class AppModule { }
```

app.routing.module.ts

```
import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';
import { AddLoansComponent } from './add-loans/add-loans.component';
import { LoanTypesComponent } from './loan-types/loan-types.component';
import { LoansComponent } from './loans/loans.component';

/*
const routes: Routes = [
  // we will create all routes here
  {
    path: 'loans',
    component: LoansComponent
  },
  {
    path: 'loans/add-loan',
    component: AddLoansComponent
  },
  {
    path: 'loans-types',
    component: LoanTypesComponent
  }
];
 */
```

```
/*
const routes: Routes = [
  // we will create multiple routes here
  {
    path: 'loans',
    component: LoansComponent
  },
  {
    path: 'add',
    component: AddLoansComponent,
    outlet:'addLoan'
  },
  {
    path: 'types',
    component: LoanTypesComponent,
    outlet:'editLoan'
  }
];
 */


const routes: Routes = [
  // we will create all routes here

  {
    path: 'loan-types',
    // component: LoanTypesComponent, // outlet:'addLoan',
    // if you write children then don't write component here
    children :[
      {
        path:'',
        component:LoanTypesComponent
      // now    http://localhost:4200/app1#/loan-types   will work
      },

      {
        path:'add-loans',
        component:AddLoansComponent
      // now    http://localhost:4200/app1#/loan-types/add-loans   will work
      },
      {
        path:'loans',
        component:LoansComponent
      // now    http://localhost:4200/app1#/loan-types/loans   will work
      },
      {
        path:'delete-loans',
        redirectTo:'loans'
      // now it ts  redirected to   http://localhost:4200/app1#/loan-types/loans
      }
    ]
  },
]
```

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

app.component.html

```
<h1 class="c1">{{title}}</h1>

<h4> Angular Chaining Pipes </h4>

<div> date Pipe :- {{ dateExampIe | date | uppercase }}</div>
<div> currencyExample Pipe :- CANADIAN  DOLLARS :- {{ currencyExample | currency :
'CAD' | lowercase}} </div>

<!-- HERE EVERY COMPONENT WILL SHOW ON ROUTER -->
<router-outlet></router-outlet>
<!-- primary outlet -->

<router-outlet name="addLoan"></router-outlet>
<!-- named router outlet -->

<router-outlet name="editLoan"></router-outlet>
<!-- named router outlet -->
```

app.component.ts

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

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  // template: `<h1>{{title}}</h1>`,
  styleUrls: ['./app.component.scss']
})

export class AppComponent {

  title = 'simpleCRM';

  userName = ""

  lowerCaseExamp1e = "ARC TUTORIALS";
  upperCaseExamp1e = "learn ANgular framework tutorials";
  dateExampIe = Date.now();
  jsonExamp1e = {username: "arc" , major: "Angular" , experience : "lem"}
```

```
  currencyExample = 125
  percentExample = 0.6767
}
```

loans.component.html

```
<p>loans works!</p>
```

loans.component.ts

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

@Component({
  selector: 'app-loans',
  templateUrl: './loans.component.html',
  styleUrls: ['./loans.component.scss']
})
export class LoansComponent implements OnInit {

  constructor() { }

  ngOnInit(): void {
  }

}
```

add-loans.component.html

```
<p>add-loans works!</p>
```

add-loans.component.ts

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

@Component({
  selector: 'app-add-loans',
  templateUrl: './add-loans.component.html',
  styleUrls: ['./add-loans.component.scss']
})
export class AddLoansComponent implements OnInit {
  constructor() { }
  ngOnInit(): void {
  }
}
```

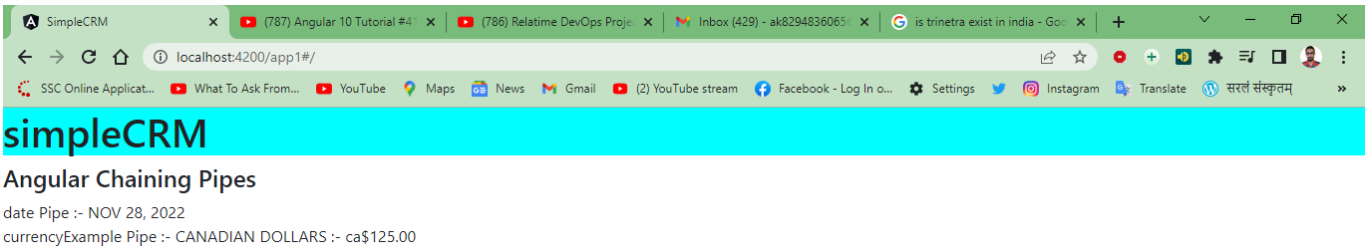
loan-types.component.html

```
<p>loan-types works!</p>
```

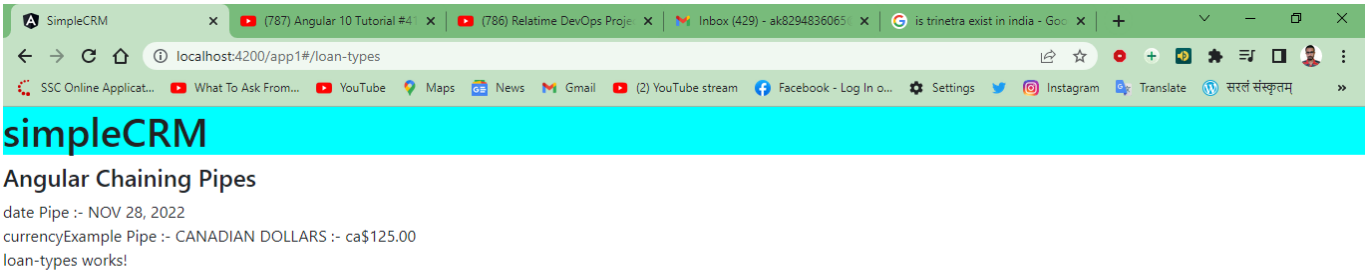
loan-types.component.ts

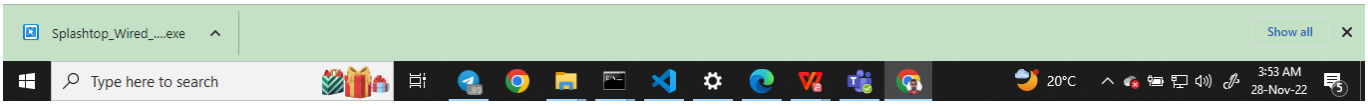
```
import { Component, OnInit } from '@angular/core';

@Component({
  selector: 'app-loan-types',
  templateUrl: './loan-types.component.html',
  styleUrls: ['./loan-types.component.scss']
})
export class LoanTypesComponent implements OnInit {

  constructor() { }

  ngOnInit(): void {
  }

}
```

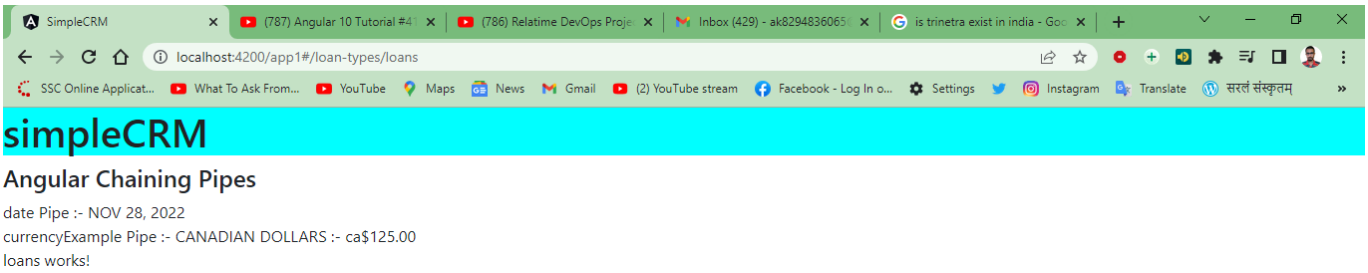
- Example:-
  - http://localhost:4200/app1#/
  - http://localhost:4200/app1#/loan-types
  - http://localhost:4200/app1#/loan-types/loans
  - http://localhost:4200/app1#/loan-types/add-loans)

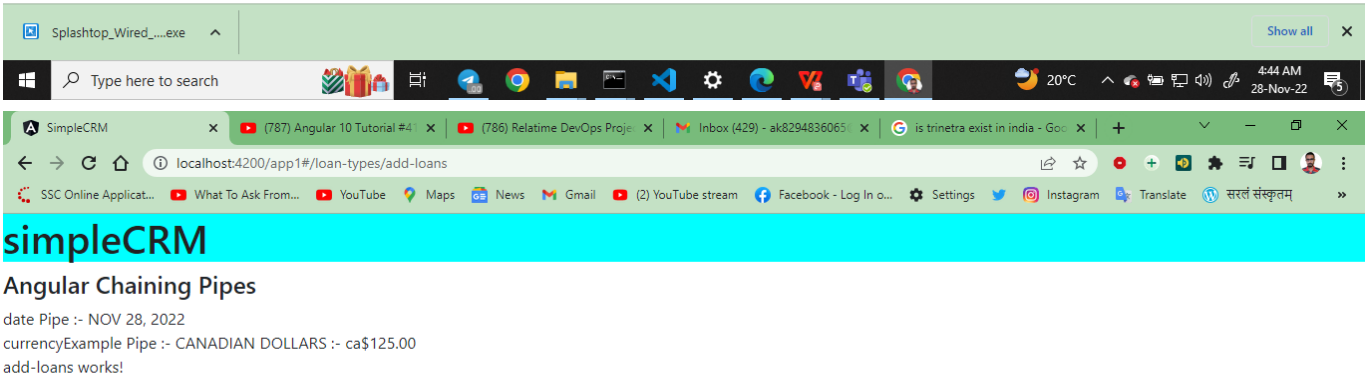

SimpleCRM
localhost:4200/app1#/
simpleCRM
Angular Chaining Pipes
date Pipe :- NOV 28, 2022
currencyExample Pipe :- CANADIAN DOLLARS :- ca$125.00

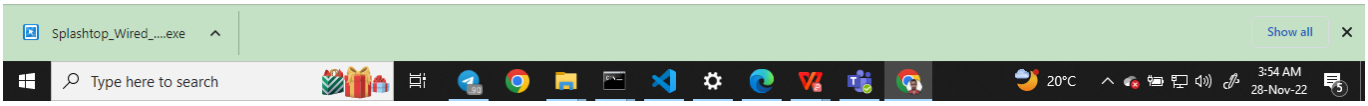

Splashtop_Wired_....exe
Show all
Type here to search
20°C
4:13 AM
28-Nov-22


SimpleCRM
localhost:4200/app1#/loan-types
simpleCRM
Angular Chaining Pipes
date Pipe :- NOV 28, 2022
currencyExample Pipe :- CANADIAN DOLLARS :- ca$125.00
loan-types works!


Splashtop_Wired_....exe
Show all
Type here to search
20°C
3:53 AM
28-Nov-22







Configure Component Routes by Angular.io

Parametrized Routes in Angular

Parametrized Routes by ARC

Parametrized Routes

We can configure and send parameters to our routes

We need to configure the route and mention that the value is dynamic `{ path :'product/:id', component: 'ComponentName'}`

- For e.g
  - product/10
  - Product/10/20

We can read the values in the component class and process the parameters

important notes :-

1. We can send dynamic data or parameters
2. URLs will look something like this
   - http://localhost.com/user/10 -> get the user with Id as 10
   - http://localhost.com/search/ka/bangalore -> state and city
   - http://localhost.com/user/10/photos/34 -> user id = 10 and photo id = 34
3. While writing dynamic `URLs/Params` - make sure you write `:(colon)` for dynamic data
4. Import the `ActivatedRoute` class
5. Create an object in `constructor` ->
6. We can create any number of dynamic params in our URLs

example -

in CLI

```
D:\theory\Angular\ANGULAR BY ARC\project\simpleCRM>ng g c p1
CREATE src/app/p1/p1.component.html (17 bytes)
CREATE src/app/p1/p1.component.spec.ts (571 bytes)
CREATE src/app/p1/p1.component.ts (260 bytes)
CREATE src/app/p1/p1.component.scss (0 bytes)
UPDATE src/app/app.module.ts (1386 bytes)

D:\theory\Angular\ANGULAR BY ARC\project\simpleCRM>ng g c p2 --module=app
CREATE src/app/p2/p2.component.html (17 bytes)
CREATE src/app/p2/p2.component.spec.ts (571 bytes)
CREATE src/app/p2/p2.component.ts (260 bytes)
CREATE src/app/p2/p2.component.scss (0 bytes)
UPDATE src/app/app.module.ts (1452 bytes)

D:\theory\Angular\ANGULAR BY ARC\project\simpleCRM>ng g c p3 --module=app
CREATE src/app/p3/p3.component.html (17 bytes)
CREATE src/app/p3/p3.component.spec.ts (571 bytes)
CREATE src/app/p3/p3.component.ts (260 bytes)
CREATE src/app/p3/p3.component.scss (0 bytes)
UPDATE src/app/app.module.ts (1518 bytes)
```

```
D:\theory\Angular\ANGULAR BY ARC\project\simpleCRM>ng g c p4 --module=app
CREATE src/app/p4/p4.component.html (17 bytes)
CREATE src/app/p4/p4.component.spec.ts (571 bytes)
CREATE src/app/p4/p4.component.ts (260 bytes)
CREATE src/app/p4/p4.component.scss (0 bytes)
UPDATE src/app/app.module.ts (1584 bytes)

D:\theory\Angular\ANGULAR BY ARC\project\simpleCRM>ng g c product --module=app
CREATE src/app/product/product.component.html (22 bytes)
CREATE src/app/product/product.component.spec.ts (606 bytes)
CREATE src/app/product/product.component.ts (280 bytes)
CREATE src/app/product/product.component.scss (0 bytes)
UPDATE src/app/app.module.ts (1670 bytes)

D:\theory\Angular\ANGULAR BY ARC\project\simpleCRM>
```

app.routing.module.ts

```
import { CommonModule } from '@angular/common';
import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';
import { AddLoansComponent } from './add-loans/add-loans.component';
import { LoanTypesComponent } from './loan-types/loan-types.component';
import { LoansComponent } from './loans/loans.component';
import { P1Component } from './p1/p1.component';
import { P2Component } from './p2/p2.component';
import { P3Component } from './p3/p3.component';
import { P4Component } from './p4/p4.component';
import { ProductComponent } from './product/product.component';

const routes: Routes = [
  { path:'product/:id' , component:ProductComponent}
  /* now any
    http://localhost:4200/app1#/product/1
    http://localhost:4200/app1#/product/2
    http://localhost:4200/app1#/product/3
    http://localhost:4200/app1#/product/4
    http://localhost:4200/app1#/product/5
    ....................................
    http://localhost:4200/app1#/product/n

    will show ProductComponent

    if we not create any object inside constructor  in product.component.ts

  */

];
```

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

app.module.ts

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

import { AppRoutingModule } from './app-routing.module';  //  importing routing
module
import { AppComponent } from './app.component';
import { HighlightDirective } from './highlight.directive';
import { UsersComponent } from './users/users.component';
import { FormsModule } from '@angular/forms';
import { LoansComponent } from './loans/loans.component';
import { LoanTypesComponent } from './loan-types/loan-types.component';
import { AddLoansComponent } from './add-loans/add-loans.component'
import { HashLocationStrategy, LocationStrategy , PathLocationStrategy } from
'@angular/common';
import { P1Component } from './p1/p1.component';
import { P2Component } from './p2/p2.component';
import { P3Component } from './p3/p3.component';
import { P4Component } from './p4/p4.component';
import { ProductComponent } from './product/product.component'; // import routing
strategy here

@NgModule({
  declarations: [
    AppComponent,
    HighlightDirective,
    UsersComponent,
    LoansComponent,
    LoanTypesComponent,
    AddLoansComponent,
    P1Component,
    P2Component,
    P3Component,
    P4Component,
    ProductComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModule, // routing module add in  imports array
    FormsModule,
```

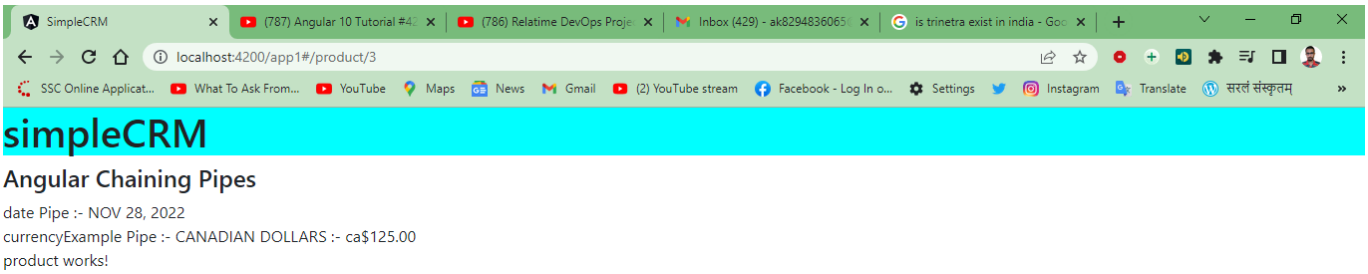
```
  ],
  providers: [
    {provide: LocationStrategy , useClass:HashLocationStrategy} , // add routing
strategy in providers array
    // {provide: LocationStrategy , useClass:PathLocationStrategy} // add routing
strategy in providers array
  ],

  bootstrap: [AppComponent]
})
export class AppModule { }
```

index.html

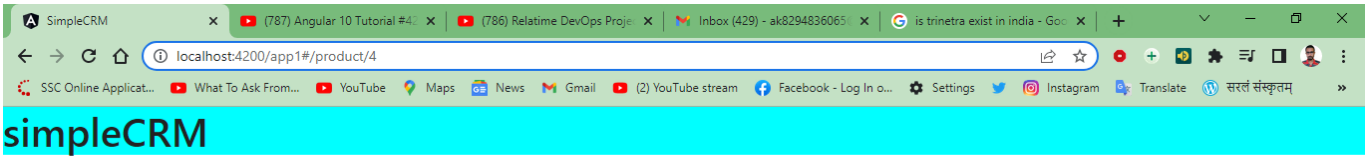
```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>SimpleCRM</title>
  <!-- <base href="/"> -->
  <base href="/app1">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="icon" type="image/x-icon" href="favicon.ico">
</head>
<body>
  <app-root></app-root>
</body>
</html>
```

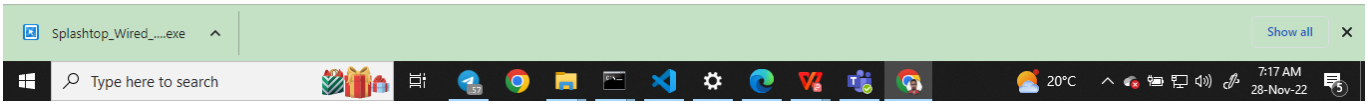
product.component.ts

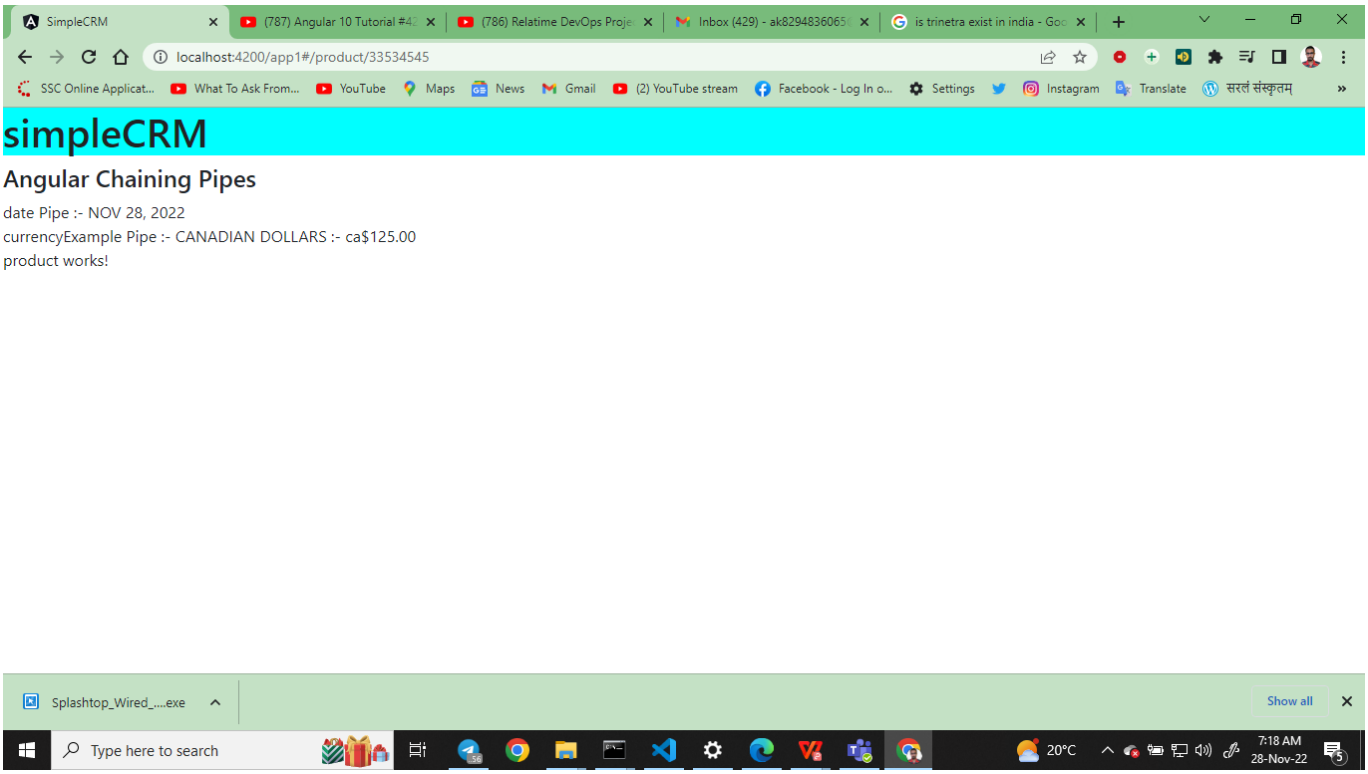
```
import { Component, OnInit } from '@angular/core';

@Component({
  selector: 'app-product',
  templateUrl: './product.component.html',
  styleUrls: ['./product.component.scss']
})
export class ProductComponent implements OnInit {

  constructor() { }

  ngOnInit(): void {
  }

}
```


SimpleCRM
localhost:4200/app1#/product/3
simpleCRM
Angular Chaining Pipes
date Pipe :- NOV 28, 2022
currencyExample Pipe :- CANADIAN DOLLARS :- ca$125.00
product works!


Splashtop_Wired_....exe
Show all
Type here to search
20°C
7:18 AM
28-Nov-22


SimpleCRM
localhost:4200/app1#/product/4
simpleCRM
Angular Chaining Pipes
date Pipe :- NOV 28, 2022
currencyExample Pipe :- CANADIAN DOLLARS :- ca$125.00
product works!


Splashtop_Wired_....exe
Show all
Type here to search
20°C
7:17 AM
28-Nov-22



example 2 :-

app.routing.module.ts

```
import { CommonModule } from '@angular/common';
import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';
import { AddLoansComponent } from './add-loans/add-loans.component';
import { LoanTypesComponent } from './loan-types/loan-types.component';
import { LoansComponent } from './loans/loans.component';
import { P1Component } from './p1/p1.component';
import { P2Component } from './p2/p2.component';
import { P3Component } from './p3/p3.component';
import { P4Component } from './p4/p4.component';
import { ProductComponent } from './product/product.component';


/*
const routes: Routes = [
  // we will create all routes here
  {
    path: 'loans',
    component: LoansComponent
  },
  {
    path: 'loans/add-loan',
    component: AddLoansComponent
  },
  {
    path: 'loans-types',
    component: LoanTypesComponent
```

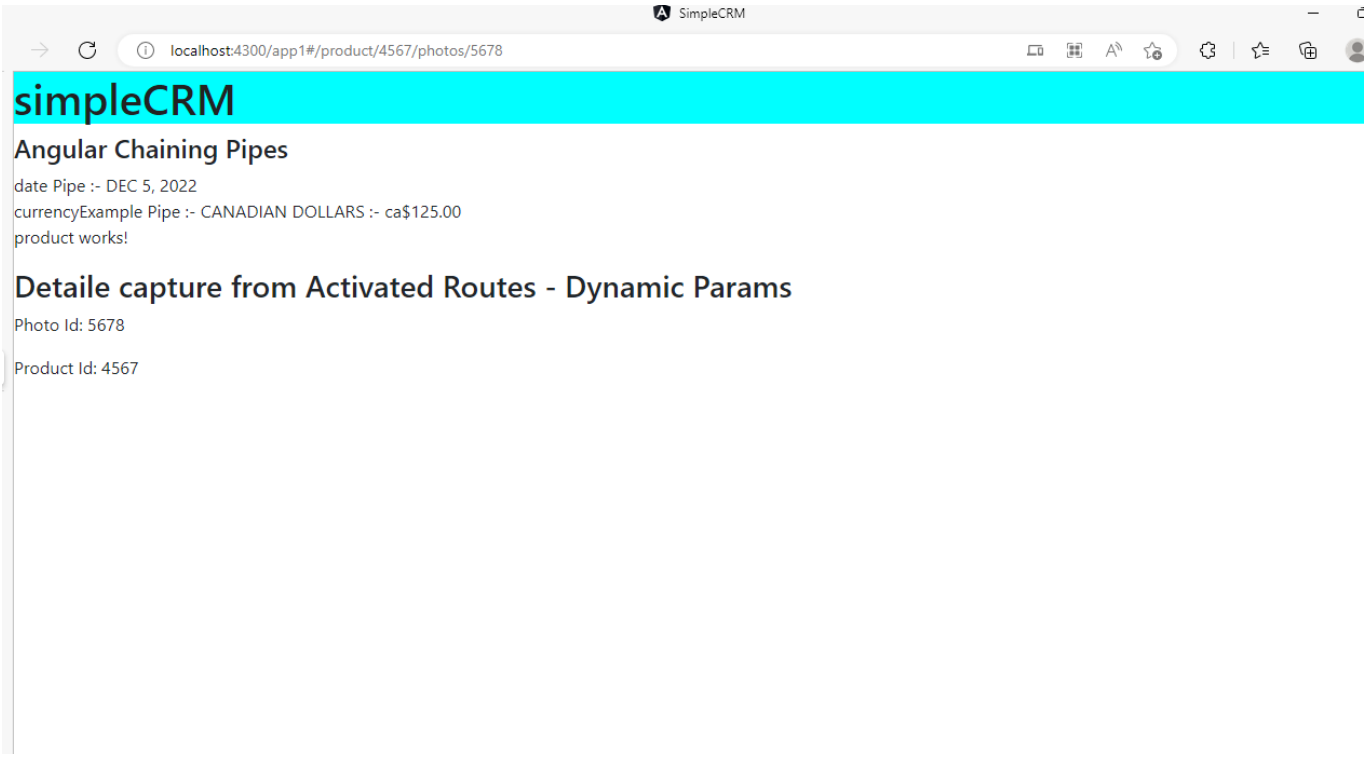
```
  }
];
 */

/*
const routes: Routes = [
  // we will create all routes here
  {
    path: 'loans',
    component: LoansComponent
  },
  {
    path: 'add',
    component: AddLoansComponent,
    outlet:'addLoan'
  },
  {
    path: 'types',
    component: LoanTypesComponent,
    outlet:'editLoan'
  }
];
 */



const routes: Routes = [

  { path:'product/:id' , component:ProductComponent} ,
  { path:'product/:productId/photos/:photoId' , component:ProductComponent}

];

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

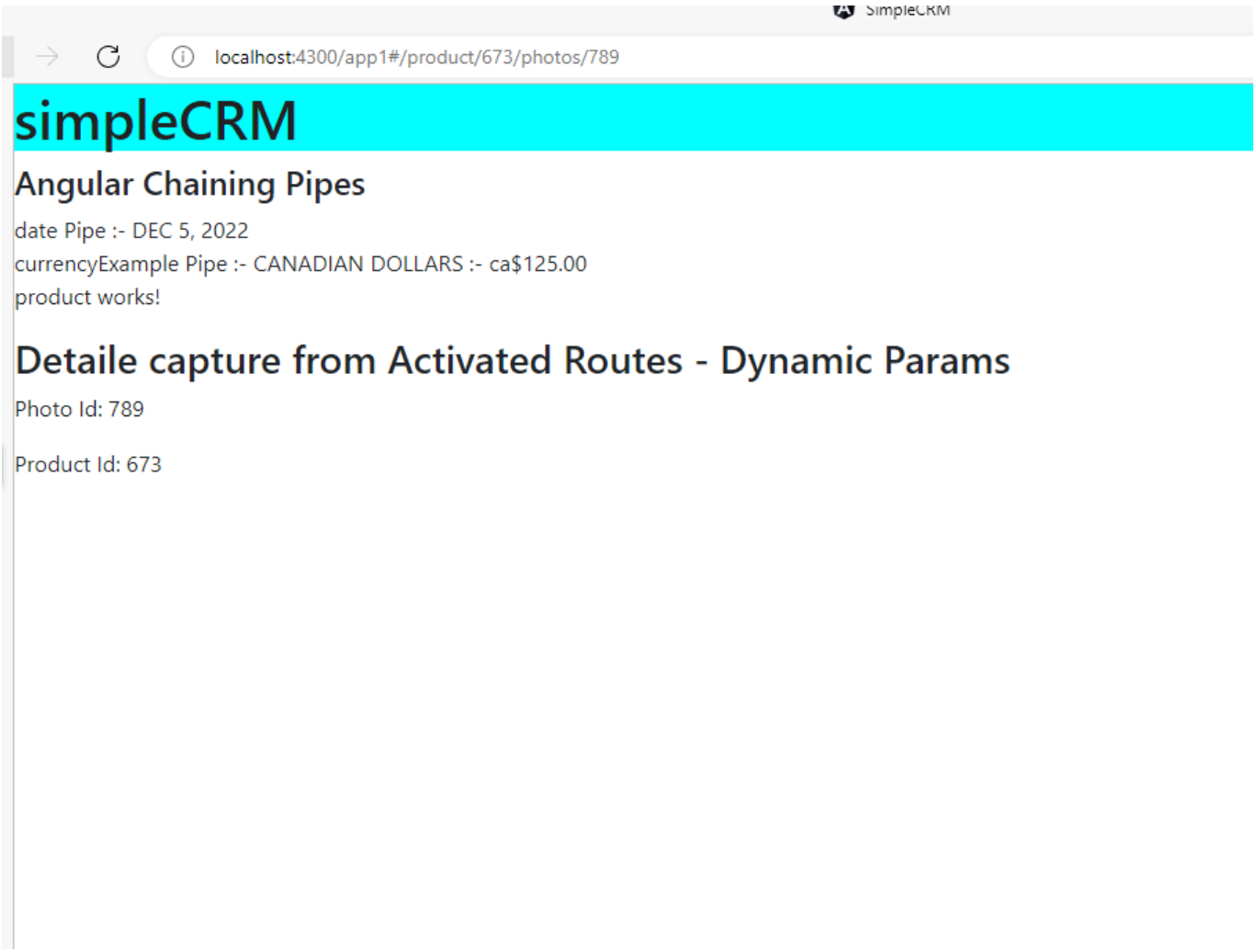
product.component.html

```
<p>product works!</p>

<h3>Detaile capture from Activated Routes - Dynamic Params</h3>

<p>Photo Id:    {{photoId}}</p>
<p>Product Id:  {{productId}}   </p>
```

product.component.ts

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

// import a class ActivatedRoute form '@angular/router'
import { ActivatedRoute } from '@angular/router';



@Component({
  selector: 'app-product',
  templateUrl: './product.component.html',
  styleUrls: ['./product.component.scss']
})
export class ProductComponent implements OnInit {

  photoId = 0 ;
  productId = 0 ;

  // create an instance of ActivatedRoute
  // instance is nothing but construction injector
  constructor(private activatedRoute : ActivatedRoute) {
    this.activatedRoute.params.subscribe((param)=> {
      this.photoId=param['photoId']
      this.productId=param['productId']
      console.log(param);
    })
  }

  ngOnInit(): void {
  }

}
```





Parametrized Routes by Angular.io

RouterLink in Angular

RouterLink by ARC

Router Link

- When applied to an element in a template, makes that element a link that initiates navigation to a route.
- Navigation opens one or more routed components in one or more `<router-outlet>` locations on the page.
- For e.g :- `<a [routerLink]="['/user/bob']"> Some link </a>`

notes:-

Episode #43 - Router Link

1. We can have any number of router links in the template
2. Router Links can be static or can be dynamic in nature
3. Common Mistakes
    - Not putting strings in single qoute
    - Not passing dyanmic data correctly
4. Static Router Link -> <a [RouterLink]="'/user'">
5. Dynamic Router Link
6. We DO NOT have to put "/" in variables in routerLink
7. Router Link Query Params -> we will cover along with Query params in routes

example

create one component that name is "clients"

```
D:\theory\Angular\ANGULAR BY ARC\project\simpleCRM>ng g c clients --module=app
CREATE src/app/clients/clients.component.html (22 bytes)
CREATE src/app/clients/clients.component.spec.ts (606 bytes)
CREATE src/app/clients/clients.component.ts (280 bytes)
CREATE src/app/clients/clients.component.scss (0 bytes)
UPDATE src/app/app.module.ts (1756 bytes)

D:\theory\Angular\ANGULAR BY ARC\project\simpleCRM>
```

app.routing.module.ts

```
import { CommonModule } from '@angular/common';
import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';
import { AddLoansComponent } from './add-loans/add-loans.component';
```

```
import { ClientsComponent } from './clients/clients.component';
import { LoanTypesComponent } from './loan-types/loan-types.component';
import { LoansComponent } from './loans/loans.component';
import { P1Component } from './p1/p1.component';
import { P2Component } from './p2/p2.component';
import { P3Component } from './p3/p3.component';
import { P4Component } from './p4/p4.component';
import { ProductComponent } from './product/product.component';



const routes: Routes = [

  // {
  //   path:'product/:id' ,
  //   component:ProductComponent
  // } ,
  // {
  //   path:'product/:productId/photos/:photoId' ,
  //   component:ProductComponent
  // } ,
  {
    path:'clients',
    component: ClientsComponent}
];


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

client.component.ts

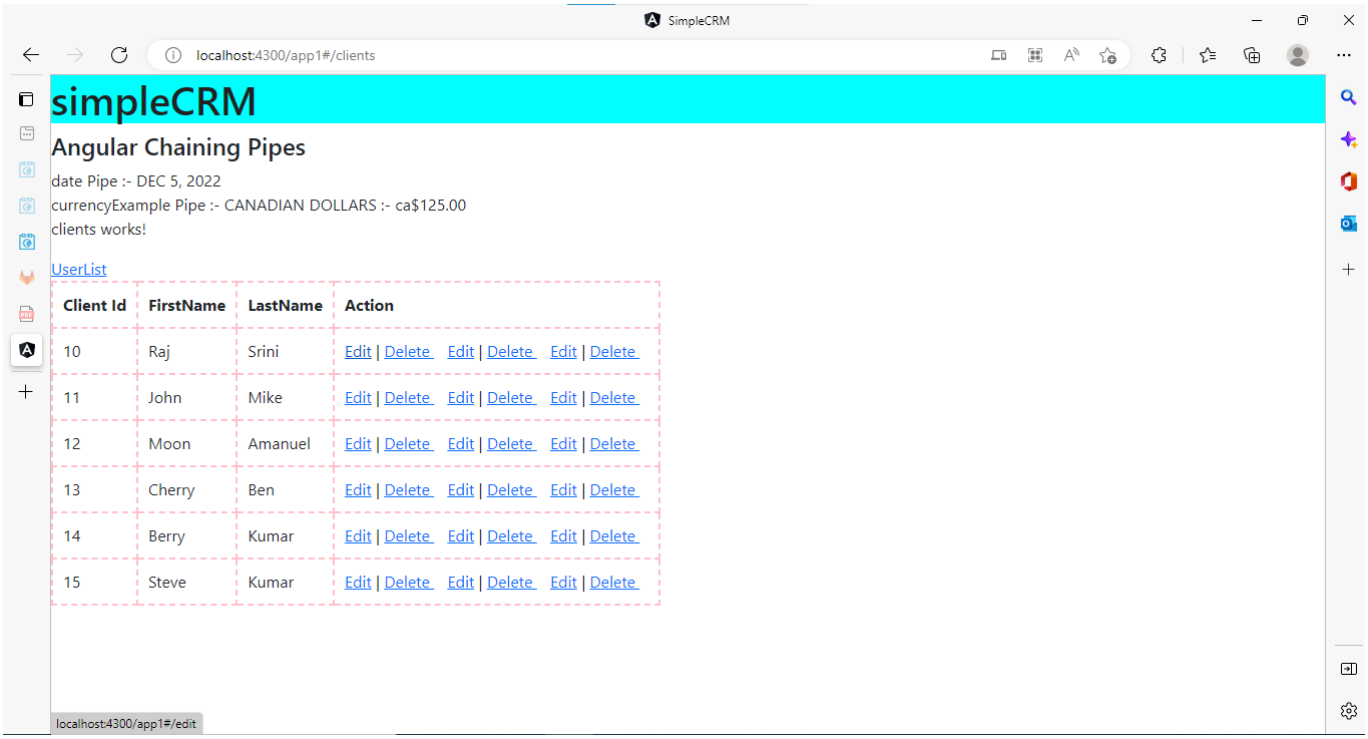
```
import { Component, OnInit } from '@angular/core';

@Component({
  selector: 'app-clients',
  templateUrl: './clients.component.html',
  styleUrls: ['./clients.component.scss']
})
export class ClientsComponent implements OnInit {

  clientList = [
    { clientId:10 , firstName : 'Raj' , lastName : 'Srini'},
    { clientId:11 , firstName : 'John' , lastName : 'Mike'},
    { clientId:12 , firstName : 'Moon' , lastName : 'Amanuel'},
    { clientId:13 , firstName : 'Cherry' , lastName : 'Ben'},
```

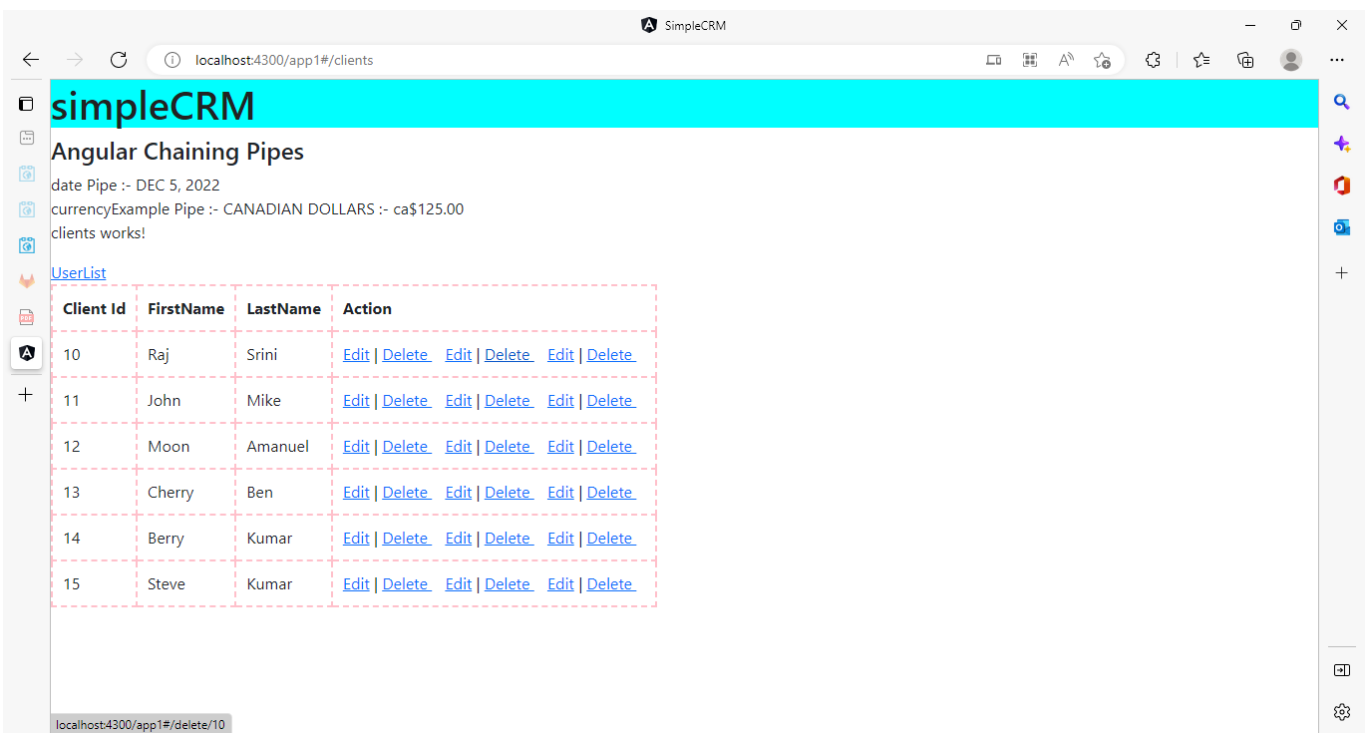
```
    { clientId:14 , firstName : 'Berry' , lastName : 'Kumar'},
    { clientId:15 , firstName : 'Steve' , lastName : 'Kumar'},
  ]

  constructor() { }

  ngOnInit(): void {
  }

}
```

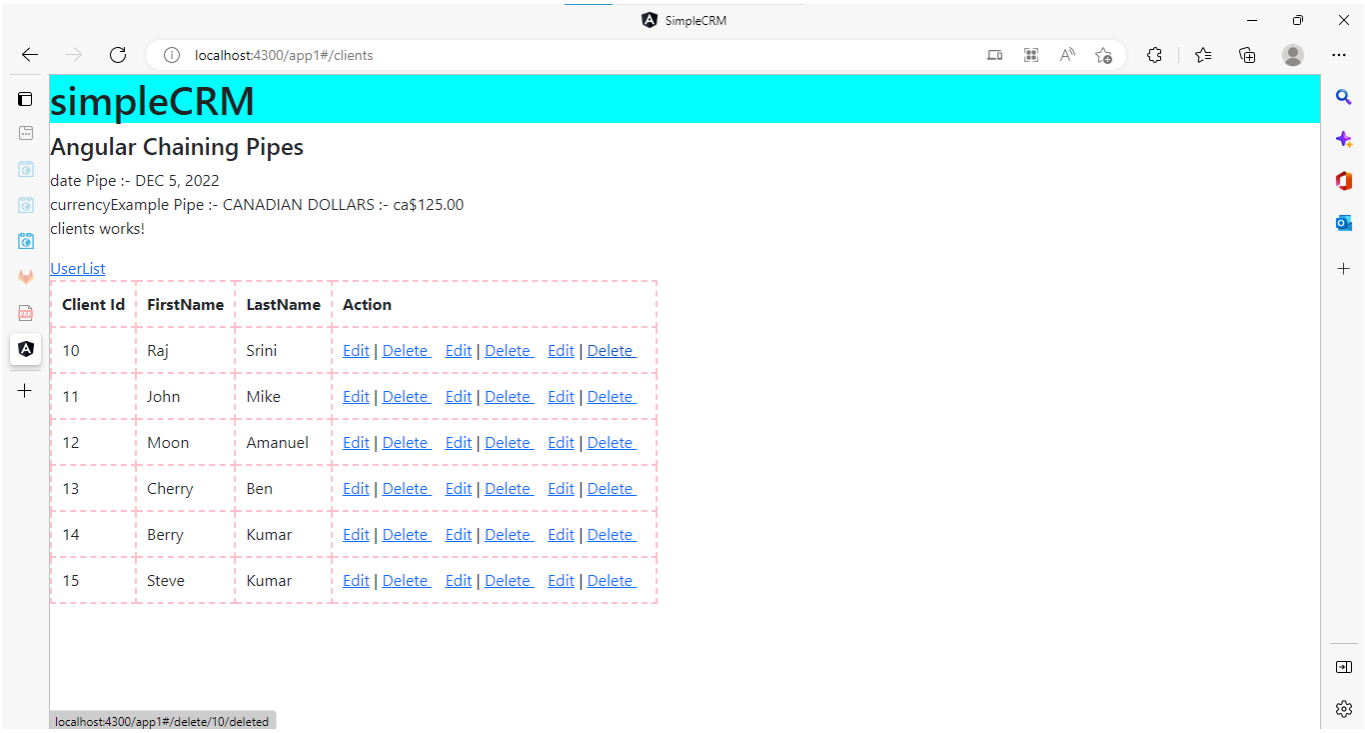
client.component.html

```
<p>clients works!</p>

<a [routerLink]="'/user'">UserList</a>

<table>
    <tr>
        <th>Client Id</th>
        <th>FirstName</th>
        <th>LastName</th>
        <th>Action</th>
    </tr>
    <tr *ngFor="let client of clientList">
        <td> {{ client.clientId }} </td>
        <td> {{ client.firstName }} </td>
        <td> {{ client.lastName }} </td>
        <td>
            <!-- when you hover on link we see output -->


            <a [routerLink]="'/edit'">Edit</a>  <!-- ti si dynamic routerlink -->
            | <a [routerLink]="'/delete'">Delete </a>   
            <!-- output :- http://localhost:4300/app1#/edit
http://localhost:4300/app1#/delete -->

            <a [routerLink]="['/edit' , client.clientId]">Edit</a>  <!-- ti si
dynamic routerlink -->
            | <a [routerLink]="['/delete' , client.clientId]">Delete </a>
  
            <!-- put '/' on only starting variable  , after we donot put '/' -->
            <!-- output :- http://localhost:4300/app1#/edit/10
http://localhost:4300/app1#/delete/10 -->
            <!-- output :- http://localhost:4300/app1#/edit/11
http://localhost:4300/app1#/delete/11 -->
            <!-- output :- http://localhost:4300/app1#/edit/12
http://localhost:4300/app1#/delete/12 -->
            <!-- output :- http://localhost:4300/app1#/edit/13
http://localhost:4300/app1#/delete/13 -->
            <!-- output :- http://localhost:4300/app1#/edit/14
http://localhost:4300/app1#/delete/14 -->
```

```
            <!-- output :- http://localhost:4300/app1#/edit/15
http://localhost:4300/app1#/delete/15 -->


            <a [routerLink]="['/edit' , client.clientId , 'edit']">Edit</a>  <!--
ti si dynamic routerlink -->
            | <a [routerLink]="['/delete' , client.clientId , 'deleted']">Delete
</a>   
            <!-- output :- http://localhost:4300/app1#/edit/10/edit
http://localhost:4300/app1#/delete/10/deleted -->
            <!-- output :- http://localhost:4300/app1#/edit/11/edit
http://localhost:4300/app1#/delete/11/deleted -->
            <!-- output :- http://localhost:4300/app1#/edit/12/edit
http://localhost:4300/app1#/delete/12/deleted -->
            <!-- output :- http://localhost:4300/app1#/edit/13/edit
http://localhost:4300/app1#/delete/13/deleted -->
            <!-- output :- http://localhost:4300/app1#/edit/14/edit
http://localhost:4300/app1#/delete/14/deleted -->
            <!-- output :- http://localhost:4300/app1#/edit/15/edit
http://localhost:4300/app1#/delete/15/deleted -->
        </td>
    </tr>
</table>
```


SimpleCRM
localhost:4300/app1#/clients
simpleCRM
Angular Chaining Pipes
date Pipe :- DEC 5, 2022
currencyExample Pipe :- CANADIAN DOLLARS :- ca$125.00
clients works!
UserList
Client Id FirstName LastName Action
10 Raj Srini Edit | Delete Edit | Delete Edit | Delete
11 John Mike Edit | Delete Edit | Delete Edit | Delete
12 Moon Amanuel Edit | Delete Edit | Delete Edit | Delete
13 Cherry Ben Edit | Delete Edit | Delete Edit | Delete
14 Berry Kumar Edit | Delete Edit | Delete Edit | Delete
15 Steve Kumar Edit | Delete Edit | Delete Edit | Delete
localhost:4300/app1#/edit


SimpleCRM
localhost:4300/app1#/clients
simpleCRM
Angular Chaining Pipes
date Pipe :- DEC 5, 2022
currencyExample Pipe :- CANADIAN DOLLARS :- ca$125.00
clients works!
UserList
Client Id FirstName LastName Action
10 Raj Srini Edit | Delete Edit | Delete Edit | Delete
11 John Mike Edit | Delete Edit | Delete Edit | Delete
12 Moon Amanuel Edit | Delete Edit | Delete Edit | Delete
13 Cherry Ben Edit | Delete Edit | Delete Edit | Delete
14 Berry Kumar Edit | Delete Edit | Delete Edit | Delete
15 Steve Kumar Edit | Delete Edit | Delete Edit | Delete
localhost:4300/app1#/delete/10



RouterLink by Angular.io

Redirect Routes in Angular

Redirect Routes by ARC

Redirecting Routes

- When we want a route to be redirected to another route — we will implement the redirectTo in our routes array
- The syntax to define the same is given below
- `{ path :", redirectTo: 'home', pathMatch: 'full'},`
- The empty path indicates that it's the **default route** of the application
- The empty path also requires us to mention that `pathMatch` should be **"full"**
- Let's learn how to redirect route in the routing module

notes :-

1. By default the root level route is ''
2. redirectTo and specify which route it has to go

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

example :-

app.routing.module.ts

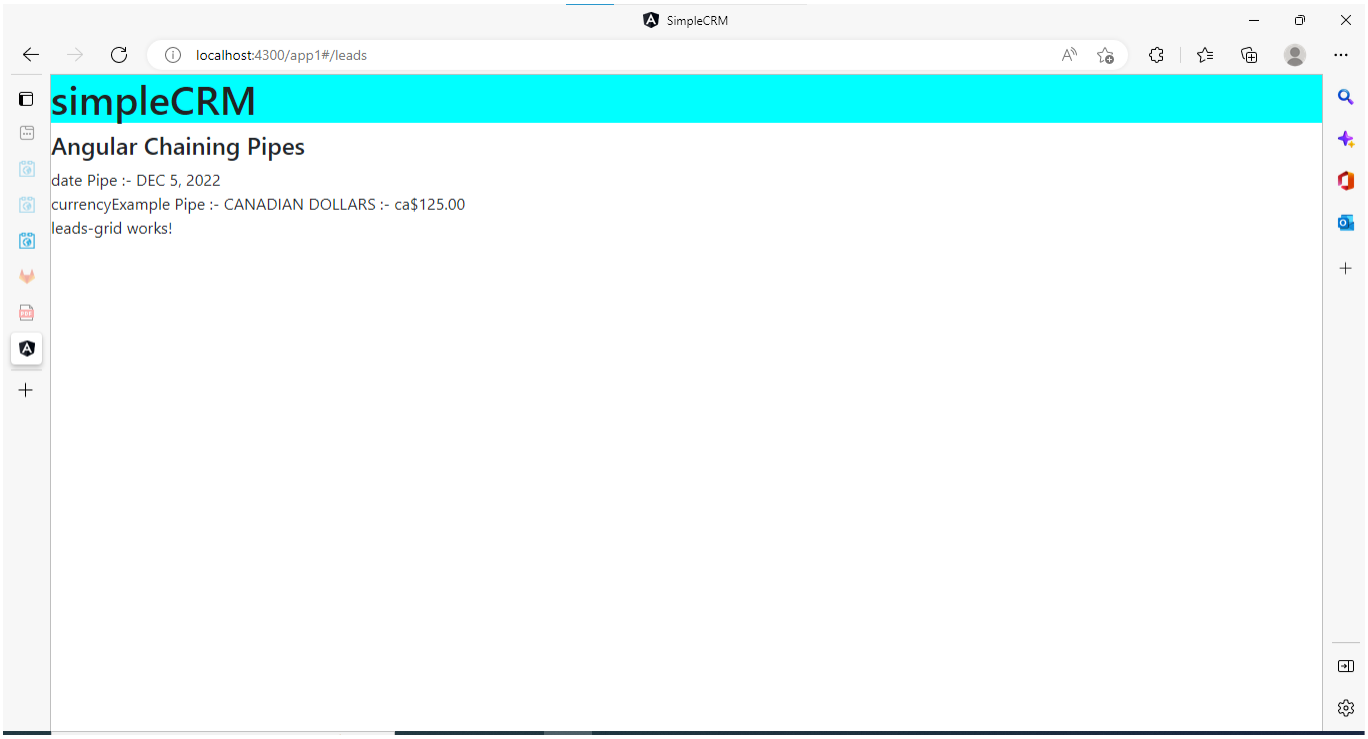
```
import { CommonModule } from '@angular/common';
import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';
import { AddLoansComponent } from './add-loans/add-loans.component';
import { ClientsComponent } from './clients/clients.component';
import { LeadsGridComponent } from './leads/leads-listing/leads-grid/leads-
grid.component';
import { LoanTypesComponent } from './loan-types/loan-types.component';
import { LoansComponent } from './loans/loans.component';
import { P1Component } from './p1/p1.component';
import { P2Component } from './p2/p2.component';
import { P3Component } from './p3/p3.component';
import { P4Component } from './p4/p4.component';
import { ProductComponent } from './product/product.component';

const routes: Routes = [
  {
    path:'clients',
    component: ClientsComponent
  },
  {
    path:'',
    redirectTo:'leads',
    pathMatch:'full'
  },
  {
    path:'leads',
    component:LeadsGridComponent
  }
];

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

leads-grid.component.html

```
<p>leads-grid works!</p>
```

leads-grid.component.ts

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

@Component({
  selector: 'app-leads-grid',
  templateUrl: './leads-grid.component.html',
  styleUrls: ['./leads-grid.component.scss']
})
export class LeadsGridComponent implements OnInit {

  constructor() { }

  ngOnInit(): void {
  }

}
```



Redirect Routes by Angular.io

Query Params in Angular

Query Params by ARC

'Query Params

- We can configure and send query parameters to our routes
- Search?keyword=toys&country=usa
- We can read the values in the component class and process the parameters

Note

1. We can send data from Form ->

2. We can have data from click ->

Basically -> URL -> http://localhost.com/search?key=10&state=ka&city=bangalore

3. Query Params -> visible in the URL

4. Most used for querying, searching or filtering data etc

facebook.com/search?page=10&pagesize=20

example :-

```
D:\theory\Angular\ANGULAR BY ARC\project\simpleCRM>ng g c profile --module=app
CREATE src/app/profile/profile.component.html (22 bytes)
CREATE src/app/profile/profile.component.spec.ts (606 bytes)
CREATE src/app/profile/profile.component.ts (280 bytes)
CREATE src/app/profile/profile.component.scss (0 bytes)
UPDATE src/app/app.module.ts (1842 bytes)

D:\theory\Angular\ANGULAR BY ARC\project\simpleCRM>ng g c search --module=app
CREATE src/app/search/search.component.html (21 bytes)
CREATE src/app/search/search.component.spec.ts (599 bytes)
CREATE src/app/search/search.component.ts (276 bytes)
CREATE src/app/search/search.component.scss (0 bytes)
UPDATE src/app/app.module.ts (1924 bytes)

D:\theory\Angular\ANGULAR BY ARC\project\simpleCRM>
```

app.routing.module.ts

```
import { CommonModule } from '@angular/common';
import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';
import { AddLoansComponent } from './add-loans/add-loans.component';
import { ClientsComponent } from './clients/clients.component';
import { LeadsGridComponent } from './leads/leads-listing/leads-grid/leads-
grid.component';
import { LoanTypesComponent } from './loan-types/loan-types.component';
import { LoansComponent } from './loans/loans.component';
import { P1Component } from './p1/p1.component';
import { P2Component } from './p2/p2.component';
import { P3Component } from './p3/p3.component';
import { P4Component } from './p4/p4.component';
import { ProductComponent } from './product/product.component';
import { SearchComponent } from './search/search.component';
```

```
const routes: Routes = [
  {
    path:'search',
    component:SearchComponent
  },
];


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

search.compnent.ts

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

@Component({
  selector: 'app-search',
  templateUrl: './search.component.html',
  styleUrls: ['./search.component.scss']
})
export class SearchComponent implements OnInit {

  color = "";
  priceTo = 0;
  priceFrom = 0;
  size = "M";


  /*
  // it is for parametrized routes
    // create an instance of ActivatedRoute
  // instance is nothing but construction injector
  constructor(private activatedRoute : ActivatedRoute) {
    this.activatedRoute.params.subscribe((param)=> {
      this.photoId=param['photoId']
      this.productId=param['productId']
      console.log(param);
    })
  }
  */


  // it is for query params
  // create an instance of ActivatedRoute
  // instance is nothing but construction injector
```

```
  constructor(private activatedRoute:ActivatedRoute) {
    this.activatedRoute.queryParams.subscribe(params=>{
      console.log(params)
      this.color = params['color']
      this.priceTo = params['priceTo'];
      this.priceFrom = params['priceFrom'];
      this.size = params['size'];
    })
   }

  ngOnInit(): void {
  }

}

/*
if i will write below link in url
http://localhost:4300/app1#/search?
id=10&color=red&size=L&priceFrom=1000&priceTo=49999

then i get output
                      Color : red
                      Size : L
                      Price From : 1000
                      Price To : 49999
*/
```

search.compnent.html

```
<p>search works!</p>

<P> Color : {{color}}</P>
<P> Size : {{size}}</P>
<P> Price From : {{priceFrom}}</P>
<P> Price To : {{priceTo}}</P>
```

app.component.html

```
<h1 class="c1">{{title}}</h1>
<router-outlet></router-outlet>  <!-- primary outlet -->
```

app.component.ts

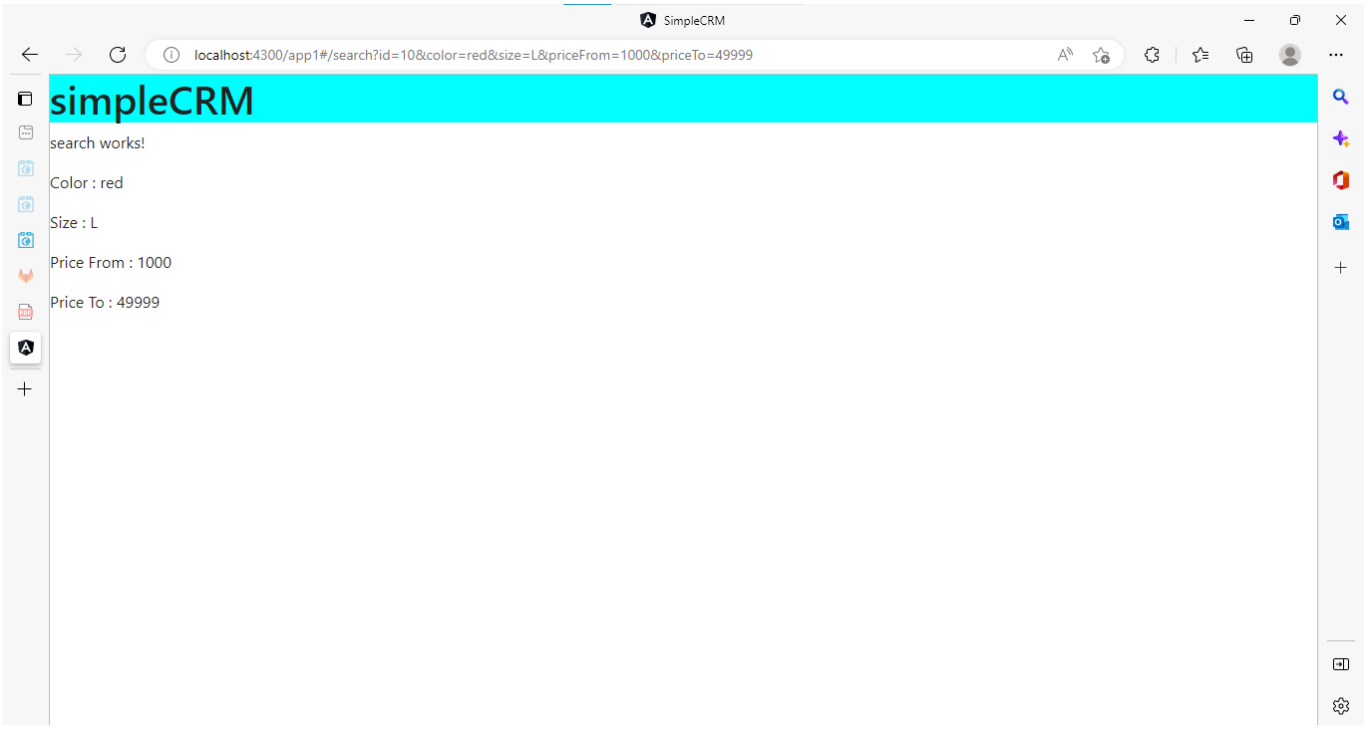
```
import { Component } from '@angular/core';

@Component({
  selector: 'app-root',
```

```
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.scss']
})

export class AppComponent {
  title = 'simpleCRM';
}
```

app.module.ts

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

import { AppRoutingModule } from './app-routing.module';  //  importing routing
module
import { AppComponent } from './app.component';
import { HighlightDirective } from './highlight.directive';
import { UsersComponent } from './users/users.component';
import { FormsModule } from '@angular/forms';
import { LoansComponent } from './loans/loans.component';
import { LoanTypesComponent } from './loan-types/loan-types.component';
import { AddLoansComponent } from './add-loans/add-loans.component'
import { HashLocationStrategy, LocationStrategy , PathLocationStrategy } from
'@angular/common';
import { P1Component } from './p1/p1.component';
import { P2Component } from './p2/p2.component';
import { P3Component } from './p3/p3.component';
import { P4Component } from './p4/p4.component';
import { ProductComponent } from './product/product.component';
import { ClientsComponent } from './clients/clients.component';
import { ProfileComponent } from './profile/profile.component';
import { SearchComponent } from './search/search.component'; // import routing
strategy here

@NgModule({
  declarations: [
    AppComponent,
    HighlightDirective,
    UsersComponent,
    LoansComponent,
    LoanTypesComponent,
    AddLoansComponent,
    P1Component,
    P2Component,
    P3Component,
    P4Component,
    ProductComponent,
    ClientsComponent,
    ProfileComponent,
    SearchComponent
  ],
  imports: [
```

```
    BrowserModule,
    AppRoutingModule, // routing module add in  imports array
    FormsModule,


  ],
  providers: [
    {provide: LocationStrategy , useClass:HashLocationStrategy} , // add routing
strategy in providers array
    // {provide: LocationStrategy , useClass:PathLocationStrategy} // add routing
strategy in providers array
  ],

  bootstrap: [AppComponent]
})
export class AppModule { }
```



Query Params by Angular.io

Wildcard Routes Link in Angular

Wildcard Routes by ARC

Wildcard Routes by Angular.io

Lazy Loading in Angular

Lazy Loading by ARC

Lazy Loading by Angular.io