

# Create A Fragment URL in Angular 8 (Jump-To-Anchor)



Rehmaanali



August 8, 2019



1800 Views

Contents [ hide ]

- 1) Router Options
- 2) Set Options forRoot
- 3) Routes
- 4) Import RoutingModule in AppModule
- 5) Create A Fragment URL in Angular 8
- 6) Fragment URL In [innerHTML]

👁 Code And Demo

The Age Of The Badass Analyst. Download The Free eBook Now.

Ad Spend Most of Your Time Uncovering Brilliant Insights. Answer Complex Quest

Alteryx

Open

In this tutorial, we will show to implement the Fragment URL in [Angular 8](#) i.e. nothing but the link which jumps to the content which contains the id mention in the fragment of an angular router.

Create Reliable Processes

Ad Automate the assignment of inbound requests to your team for faster delivery with '

Wrike

Learn more

let's create the app routing assume wich display the home page content on the `/content` url. i.e on `/content` url our main `app.component.ts` / `.html` data will be displayed .

app-routing.module.ts



```
import { NgModule } from '@angular/core';
import { CommonModule } from '@angular/common';
import { RouterModule, Routes, ExtraOptions } from '@angular/router';
import { AppComponent } from '../app.component';
const routerOptions: ExtraOptions = {
  scrollPositionRestoration: 'enabled',
  anchorScrolling: 'enabled',
  scrollOffset: [0, 64],
};
const routes: Routes = [
  {
    path: 'content',
    component: AppComponent,
  },
  {
    path: '',
    redirectTo: 'content',
    pathMatch: 'full'
  }
]
@NgModule({
  imports: [CommonModule, RouterModule.forRoot(routes, routerOptions)],
  exports: [RouterModule]
})
export class AppRoutingModule { }
```

lets split up the router module parts to make it understand properly.

### Router Options

we will create a configuration which will contain options for the router to make our fragment router attribute work.

Router Options

```
const routerOptions: ExtraOptions = {
  scrollPositionRestoration: 'enabled',
  anchorScrolling: 'enabled',
  scrollOffset: [0, 64],
};
```

**scrollPositionRestoration:** on change of router url the position of the screen will set to the top.

**anchorScrolling :** When set to ‘enabled’, scrolls to the anchor element when the URL has a fragment. Anchor scrolling is disabled by default.

**scrollOffset:** Configures the scroll offset the router will use when scrolling to an element.

### Set Options forRoot

set the **ExtraOptions** to the **RouterModule.forRoot()** method.

Router Options

```
@NgModule({
  imports: [CommonModule, RouterModule.forRoot(routes, routerOptions)], //Export
  ...
})
```

### Routes

as it is mention that we will route our home page on **/content** so the routes are declared as:

Router Options



Create Reliable Proce  
Automate the assignment of  
for faster delivery with Wrike  
Ad Wrike

```
const routes: Routes = [
  {
    path: 'content',
    component: AppComponent,
  },
  {
    path: '',
    redirectTo: 'content', // Empty path will redirect to content route.
    pathMatch: 'full'
  }
];
```

### Create Reliable Processes

Ad Automate the assignment of inbound requests to your team for faster delivery with 'Wrike'

Wrike

Learn more

## Import RouterModule in AppModule

Now just import the RouterModule file in our main app module file so it can be run thought out the application.

app.module.ts

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { AppComponent } from './app.component';
import { HelloComponent } from './hello.component';
import { RouterModule, Router } from '@angular/router';
import { AppRoutingModule } from './app-routing/app-routing.module';
@NgModule({
  imports:      [ BrowserModule, RouterModule, AppRoutingModule ], // import
  declarations: [ AppComponent, HelloComponent ],
  bootstrap:   [ AppComponent ],
  providers: [  ]
})
export class AppModule { }
```

### Create Reliable Processes

Ad Automate the assignment of inbound requests to your team for faster delivery with 'Wrike'

Wrike

Learn more

## Create A Fragment URL

At the final, we will create a markdown which will contain an angular attribute i.e routerLink and fragment.

The fragment will contain the id of that section and that ID will use in anchor tag!

Suppose we have array of content

app.component.ts



Incase if your markup data are loaded from API and that contain hash content routing i.e fragment URL in `[innerHTML]` you can manually replace some URLs to make it work in angular application.

you can see my example the way I implemented it.

blog-content.component.ts

```
import { Component } from '@angular/core';
@Component({
  selector: 'my-app',
  templateUrl: './app.component.html',
  styleUrls: [ './app.component.css' ]
})
export class BlogContentComponent implements OnInit, AfterViewChecked {
  @Input()
  blog_content: String;
  @Input()
  slug: String;
  constructor() {}
  ngOnInit() {}
  ngAfterViewChecked() {
    if (this.blog_content ) {
      const anchors = document.getElementsByTagName('a');
      for (let i = 0; i < anchors.length; i++) {
        if (anchors[i].hash) { // hash URL
          const hash = anchors[i].hash.replace('#', '');
          anchors[i].setAttribute('href', `/${this.slug}#${hash}`);
          anchors[i].setAttribute('fragment', `${hash}`);
          anchors[i].setAttribute('ng-reflect-fragment', `${hash}`);
          anchors[i].setAttribute('routerlink', `/${this.slug}`);
          anchors[i].setAttribute('ng-reflect-router-link', `/${this.slug}`);
        } else if (anchors[i].href.match(environment.site_url_regex)) { // site url
          const url = anchors[i].href.replace(environment.site_url, ''); // site url
          anchors[i].setAttribute('href', `${url}`);
          anchors[i].setAttribute('routerlink', `${url}`);
          anchors[i].setAttribute('ng-reflect-router-link', `${url}`);
        }
      }
    }
  }
}
```

this will conver normal URL to angular router links with fragments.

and html will be same as it is

blog-content.component.html

```
<article class="article" [innerHTML]="blog_content | safeHtml"></article>
```

Find More Articles on [Angular](#).

Share this:

Twitter

Facebook

Reddit

LinkedIn

WhatsApp

Print

Tumblr

Related



Routing In AngularJS Using UI-Router

June 18, 2017



Best Ways To Preload modules In Angular 9

June 2, 2020



Router Change Events In Angular 8+ Router Class

March 4, 2020

Tags: [Angular 8](#), [Angular Router](#), [Fragment Url](#), [routes](#)

```
import { Component } from '@angular/core';
@Component({
  selector: 'my-app',
  templateUrl: './app.component.html',
  styleUrls: [ './app.component.css' ]
})
export class AppComponent {
  name = 'Angular';
  ids: Array<String> = ['one', 'two', 'three', 'four']
}
```

So in our `app.component.html` we will configure our anchor tags as mention below

app.component.html

```
<div class="links">
  <ul>
    <li *ngFor="let link of ids">
      <a [routerLink]='."' [fragment]="link">{{link}}</a>
    </li>
  </ul>
</div>
```

and the section with `id` attribute will be as follow

app.component.html

```
<section>
  <div class="section" *ngFor="let link of ids" [attr.id]='link' >
    <h1>{{link}}</h1>
    <div class="content">
      <!-- YOUR CONTENT GOES HERE -->
    </div>
  </div>
</section>
```

Demo

app.component.html ×

1 <router-outlet></router-outlet>

fragment-url-in-angular Editor Preview Both Edit on ⚡ Stack

Create Reliable Processes


Ad Automate the assignment of inbound requests to your team for faster delivery with '

Wrike

Learn more

Fragment URL In [innerHTML]






tryinghard

Reply

How can i change this so each paragraph is different rather than the same thing repeated over and over?




Rehmaanali

Reply

It can be based on your data for the demo I have just repeated the same content.  
your data can be like

```
[
  {id: 'asdasd-asd-asdas', post: '<p>some content...</p>'},
  {id: 'cxxfsd-scs-rded', post: '<p>some content...</p>'}
]
```


so Inside `[innerHTML]` you can pass **post** object



Anonymous

Reply

It was not helpful to me as focus is not set on target so not acting like a normal internal anchor




Rehmaanali

Reply

Hi  
You can set the page offset to your required position using route config extraOptions


Reffer: <https://angular.io/api/router/ExtraOptions>



CJRamki

Reply

scrollPositionRestoration is breaking anchorScrolling functionality. Is there anyway to fix?



Rehmaanali

Reply

hi **CJRamki**,  
You must be referring to the single page application demo due to that it was not able to route  
I have made changes in the demo you can refer the code from below link  
[Demo and Code](#)

Changes :  
`scrollPositionRestoration` will work if the `<router-outlet></router-outlet>` is in the application.

## Leave a Reply

Your email address will not be published. Required fields are marked \*

Comment

Name \*

Email \*

Website

☐ Notify me of follow-up comments by email.

☐ Notify me of new posts by email.

Post Comment

This site uses Akismet to reduce spam. [Learn how your comment data is processed.](#)





Create Reliable Processes

Automate the assignment of inbound requests to your team for faster delivery with Wrike!

Ad [Wrike](#)



✉ [Subscribe Us](#)

[Subscribe Us](#)



Geekstrick By [@rehmaanali](#)

A blog on dev explaining the tricks to make development ease.

Trending Topics

- [JavaScript](#)
- [Angular 9+](#)
- [AngularJS](#)
- [NodeJS](#)

Quick Links

- [Tutorials](#)
- [Courses](#)
- [Tech Talks](#)
- [Ask Question](#)

