Angular 2 Router Cheat Sheet

by Nathan (Nathane2005) via cheatography.com/28056/cs/8478/

Getting started

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

Router Imports

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

Sample Router

```
RouterModule.forRoot([
      { path: 'hero/:id',
component: HeroDetailComponent },
      { path: 'crisis-center',
component: CrisisListComponent },
      {
        path: 'heroes',
        component:
HeroListComponent,
        data: {
          title: 'Heroes List'
        }
      },
      { path: '', component:
HomeComponent },
      { path: '**', component:
PageNotFoundComponent }
    ])
  ]
```

There is an array of routes defined for the specific section of the application, depending on the embedded nature of the application there will be more child routes defined

General

1. The order of the routes matter
2. There is usually a default route with an empty path

Router Outlet

```
<!-- Routed views go here -->
<router-outlet></router-outlet>
```

Router Link

RouterLink	The directive for binding a clickable HTML element to a route. Clicking an anchor tag with a routerLink directive that is bound to a string or a Link Parameters Array triggers a navigation.
RouterLinkActive	The directive for adding/removing classes from an HTML element when an associated routerLink contained on or inside the element becomes active/inactive.
ActivatedRoute	A service that is provided to each route component that contains route specific information such as route parameters, static data, resolve data, global query params and the global fragment.

Router Link (cont)

Router State	The current state of the router including a tree of the currently activated routes in our application along convenience methods for traversing the route tree.
Router	Displays the application component for the active URL. Manages navigation from one component to the next.
Router Module	A separate Angular module that provides the necessary service providers and directives for navigating through application views.
Routes	Defines an array of Routes, each mapping a URL path to a component.
Route	Defines how the router should navigate to a component based on a URL pattern. Most routes consist of a path and a component type.

By **Nathan** (Nathane2005)
cheatography.com/nathane2005/

Router Controls

CanActivate	Checking route access
CanActivateChild	Checking child route access
CanDeactivate	Ask permission to discard unsaved changes
CanLoad	Check before loading feature module assets

Child Routes Definition

```
RouterModule.forChild([
      { path: 'heroes', component:
HeroListComponent },
      { path: 'hero/:id',
component: HeroDetailComponent }
    ])
```

In the parent definition include the loadChildren attributes

Lazy Loading Of Routes

```
export function
loadCompensationModule() {
  return require('es6-
promise!../cf/shell/shell.module')
('ShellModule');
}
export const routes: Routes = [
  { path: '', redirectTo: 'home',
pathMatch: 'full'},
  { path: 'cf', loadChildren:
loadCompensationModule},
  { path: 'home', component:
MainComponent, canActivate:
[AuthGuard]},
```

Lazy Loading Of Routes (cont)

```
  { path:'**', redirectTo: '' }
];
export const appRoutingProviders:
any[] = [
];
export const routing:
ModuleWithProviders =
RouterModule.forRoot(routes, {
useHash: false});
```

This is a sample. The children definition would be defined in the children array namely:

```
export const routes: Routes = [
{
path: '', component: ShellComponent,
children: [
{path: 'home', component: ShellComponent},
{path: 'one', loadChildren: loadOneModule},
{path: 'two', loadChildren: loadTwoModule},
{path: 'three', loadChildren: loadThreeModule},
{path: 'four', loadChildren: loadFourModule}
]
},
{path: '**', redirectTo: 'home'}
];
```

Route Parameters - Mandatory

Route Parameter	xxx/:{type}

To Specify the route information
1. Construct the URL
2. Pass in this.router.navigate(['/hero', hero.id]);

Route Paramters - Optional

```
<a [routerLink]="['/crisis-center',
{ foo: 'foo' }]">Crisis Center</a>
```

Location Stategy

PathLocationStrategy	the default "HTML 5 pushState" style.
HashLocationStrategy	the "hash URL" style.

RouterModule.forRoot(routes, { useHash: true }) // .../#/crisis-center/

CanActivate Guard

```
import { Injectable } from
'@angular/core';
import {
  CanActivate, Router,
  ActivatedRouteSnapshot,
  RouterStateSnapshot
} from '@angular/router';
import { AuthService } from
'./auth.service';
@Injectable()
export class AuthGuard implements
CanActivate {
  constructor(private authService:
AuthService, private router:
Router) {}
  canActivate(route:
ActivatedRouteSnapshot, state:
RouterStateSnapshot): boolean {
    let url: string = state.url;
    return this.checkLogin(url);
  }
  checkLogin(url: string): boolean
{
    if
(this.authService.isLoggedIn) {
return true; }
    // Store the attempted URL for
redirecting
    this.authService.redirectUrl =
url;
```

CanActivate Guard (cont)

```
    // Navigate to the login page with extras
    this.router.navigate(['/login']);
    return false;
  }
}
```

To reference the guard:

canActivate: [AuthGuard],

Route Animations

```
Import
import { Component, OnInit, HostBinding,
         trigger, transition, animate,
         style, state } from '@angular/core';
animations: [
    trigger('routeAnimation', [
      state('*',
        style({
          opacity: 1,
          transform: 'translateX(0)'
        })
      ),
      transition('void => *', [
        style({
          opacity: 0,
          transform: 'translateX(-100%)'
        }),
        animate('0.2s ease-in')
      ]),
      transition('* => void', [
```

Route Animations (cont)

```
        animate('0.5s ease-out', style({
          opacity: 0,
          transform: 'translateY(100%)'
        }))
      ])
    ])
  ]
  @HostBinding('@routeAnimation') get routeAnimation() {
    return true;
  }
```

the host binding is for linking to the transition effect.