**Router**

**Step 1 – Parse the URL**

How router work in Angular?

/ : Slashes divide URL segments

(): Parentheses specify secondary routes

**Step 2 – Redirect**

**There are 2 kinds of redirect**

* Local Redirect : not start with slash {path:’one’, redirectTo:’two’}
* Absolute Redirect: When redirect start with a slash {path:’one’, redirectTo:’/two’}

**Step 3 – Identify the router**

**Step 4 – run guard**

**Step 5 – Run Resolvers**

**Step 6 – Activate**

**Step 7 – Manage**

**Routing | Router Outlet**

<router-outlet></router-outlet>

**Routing | Router Link (in View)**

<a [routerLink]=”[‘/students’]”>Students</a>

**Routing | Router Link (Server Side)**

* Import {Router} from ‘@angular/router’
* Constructor(private router:Router){}
* This.router.navigate([‘/student’,73845]);

**Routing | Generate Router**

Ng generate module app-routing –flat –module=app

--flat puts the file in src/app and will not create own folder

-- module=app tells the CLI to register ot in the imports array of the AppModule.

**Routing | How Create app-routing.module via self coading?**

Import{ngModule} from ‘@angular/core’;

Import{Routers, RouterModule} from ‘@angular/router;

Const routes:Routes = [

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

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

]

@NgModule({

Imports: [RouterModule.forRoot(routes)],

Exports:[RouterModule]

})

Export class AppRoutingModule {}

**Routing | Redirecting route**

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

Note:   
- This line come on very first in routing array function.  
- We also use “pathMatch:’full’” property with redirectTo property other wise it show error.

**Routing | Wildcard Route in**

{path:’\*\*’, component:PageNotFoundComponent}

Use wildcard route to intercept invalid URLS and handle it gracefully.

**Routing | Children Route in**

{

path:’pagename’,

children:[

{path:’’, component: ParentpageComponent },

{path:’children-pagename1’, component: ChildrenComponent },

{path:’children-pagename1’, component: ChildrenComponent }

]

}

* First path keep blank and set parent page component for load parent pate
* Links from parent page like below:  
  <a [routerLink]=[‘./clild-page’]>Child Page</a>
* Link from child to siblings will like below:  
  <a [routerLink]=[‘../chile-page2’]>Child Page2</a>