<!-- MỘT SỐ LƯU Ý KHI LÀM VIỆC VỚI ROUTER , COMPONENT TRONG ANGULAR  
  
0. Muốn sử dụng componet này vào component khác để giảm thiểu code:  
 B1: ta tạo ra 1 component để sử dụng chung (thường là header , footer)  
 B2: trong componet khác muốn gọi vào thì sử dựng với cú pháp là <app-footer></app-footer> hoặc <app-header></app-header>  
 :: lưu ý app-footer hay app-hearder là tên nằm trong file ....+ component.ts ở vị trí selector ta có thể thay đổi tên này đc  
  
  
1. Ta đứng ở Component trong một cái module con do ta tạo ra ,  
mà muốn sử dụng 1 component độc lập (Nghĩa là component ta đang muốn sử dụng  
nó không nằm trong bất kĩ module nào do ta tạo ra)  
thì ở app-routing.module (cha) phải sử dụng trong :  
const routes: Routes = [  
 {path: "đăt tên link component muốn lấy để sd", component: Tên component mà ta muốn lấy để sử dụng}  
 ]  
=> khi đó ở vị trí ta đang đứng để thực hiện (component trong module con),  
trong file html sử dụng routerLink tại thẻ a với cú pháp:  
routerLink="tên link component muốn lấy để sd " . Phải giống tên ở phần path ở app-routing.module (cha)  
  
vd thực tế: sử dụng component đk tài khoản (nằm độc lập ko nằm trong module nào cả ) ở trong cái module security có component là login , reset password ,....  
và mục tiêu là : ở tại component login (nằm trong module con) tại file html ở component login muốn gọi giao diện component đk tài khoản  
  
  
2. Nếu muốn sử dụng qua lại các component trong module con do ta tao ra  
bằng cách sử dụng thẻ a và href="đăt tên link component muốn sd" có thể sử dung routerLink= "/.. " thì  
cấu hình trong file routing của module con như sau:  
 const routes: Routes = [  
 {path :'đăt tên link component muốn sd' , component: tên component muốn sử dụng}  
 ];  
  
 => khi đó tại vị trí thẻ a trong component mà ta muốn sd, thực hiện cú pháp href = "/tên link component muốn sd"  
  
vd : trong module con do ta tạo có component : về Information , Quản lý đơn đặt hàng ,... ,  
 Và ta muốn chuyển đổi qua lại giữa các component này  
  
  
3. sử dụng giữa các module : module cha load module con:  
 B1: Tại app-routing.module (cha) thực hiện cú pháp:  
 const routes: Routes = [  
 {  
 path : "ten do ta quy dịnh trước khi vào từng component trong module con", canActivate: [AuthGuardService],  
 loadChildren:() => import('./component/ten-module/ten module + .module').then(module =>module.ten module + Module)  
 },...tương tự các module khác....  
 ]  
  
 B2: Trong file routing của module con sử dụng router bình thường  
 const routes: Routes = [  
 {  
 path: "ten dat cho component con trong module con", component: ten component con  
 },...  
 ];  
  
 B3: Sử dụng:  
 Tại thẻ a dùng với cú pháp như sau : routerLink="/tên path ở component cha/tên path ở componet con"  
  
  
 Vd thực tế : thường người ta chia module thể thực hiện cơ che lazy load (nghĩa là load từng module )  
 và thường chia ra các module nhỏ để làm teamrork  
  
  
:: Lưu ý chung : tất cả khi cấu hình path ở router ko có kí tự "/"  
 nhưng khi sử dụng ở thẻ a để chuyển đổi qua lại giữa các component thì sử dụng "/"

4. Khi sử dụng header , footer làm module chung thì ở module nào sử dụng heafer và footer thì ta import header và footer module vào module mà ta cần sử dung header và footer.  
  
-->