
 Se former autrement	Année Universitaire : 2022-2023 
Workshop n°5 : <h2 style="text-align: center;">Lazy loading</h2>	

### Objectifs :

- Créer des Modules à part les le module racine.
- Attacher les composants aux modules correspondants
- Appliquer le lazy loading sur un projet existant.

### *Etat initial de l'application :*

Supposons que nous avons l'application Angular décrite ci-dessous :

- La hiérarchie de ses composants est illustrée par la figure 1.
- En cliquant sur le lien :
  - Home, le composant HomeComponent est affiché
  - UserManagement, le composant UserComponent est affiché
  - ProductManagement, le composant ProductComponent est affiché
  - ProviderManagement, le composant ProviderComponent est affiché
- En tapant dans l'url un path non valide, le composant NotFoundComponent est affiché
- Par défaut (localhost :4200), le composant HomeComponent est affiché
- La liste des composants attachés au module racine est décrite par la figure 2.

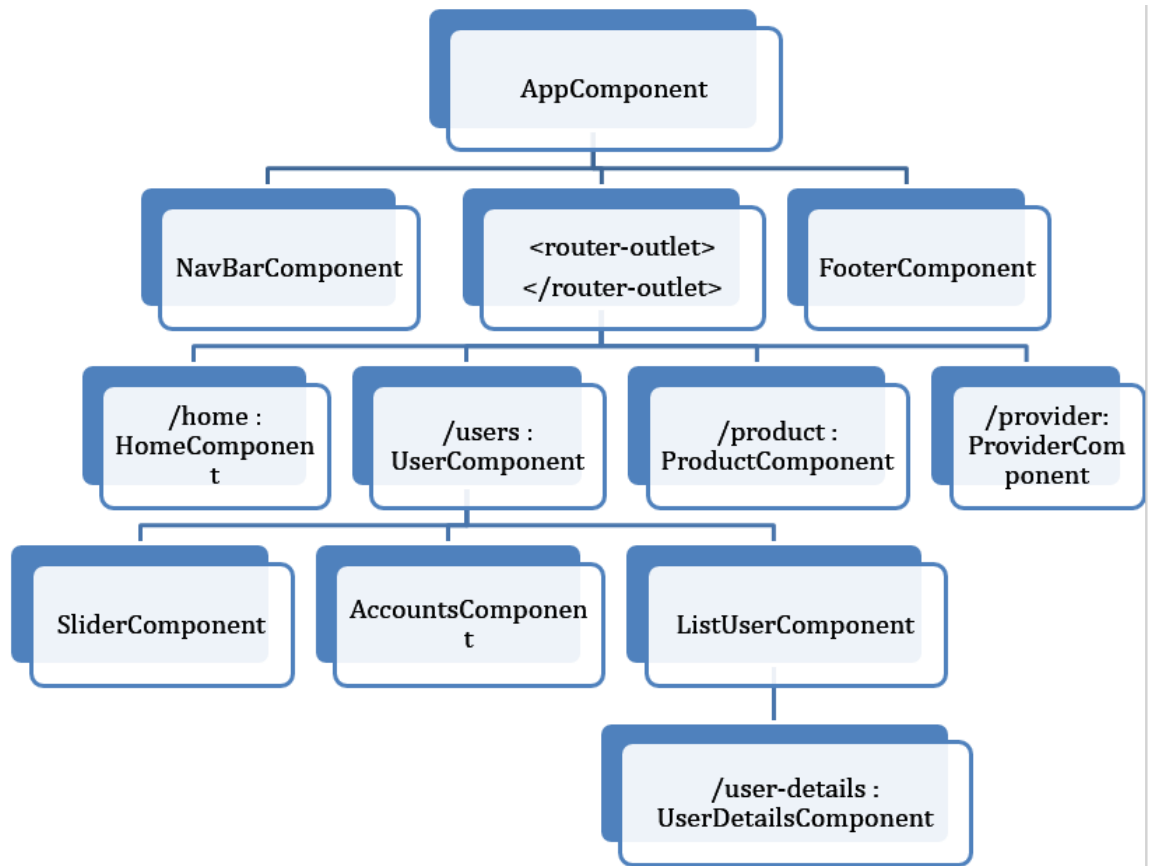


Figure 1

```

@NgModule({
  declarations: [
    AppComponent,
    ProductComponent,
    NavBarComponent,
    SliderComponent,
    AccountsComponent,
    ListUserComponent,
    FooterComponent,
    HomeComponent,
    UserComponent,
    UserDetailsComponent,
    NotFoundComponent,
    ProviderComponent
  ],

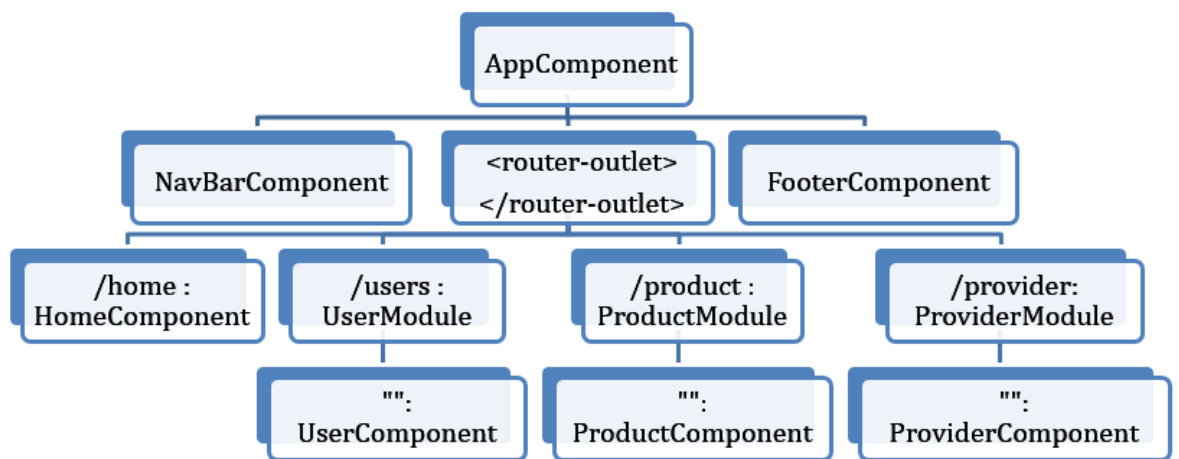
```

Figure 2

### ***Etat souhaité de l'application :***

Nous voulons avoir la hiérarchie de l'application décrite par la figure 3

- En cliquant sur le lien :
  - Home, le composant HomeComponent est affiché
  - UserManagement, le module UserModule est chargé et le composant UserComponent est affiché
  - ProductManagement, le module ProductModule est chargé et le composant ProductComponent est affiché
  - ProviderManagement, le module ProviderModule est chargé et le composant ProviderComponent est affiché
- En tapant dans l'url un path non valide, le composant NotFoundComponent est affiché
- Par défaut (localhost :4200), le composant HomeComponent est affiché



**Figure 3**

### ***Travail à faire :***

- 1- Créer les modules UserModule, ProductModule, ProviderModule avec leurs modules de routage. Exemple, pour générer le module UserModule avec son module de routage.

```
ng g m User --routing
```

- 2- Attacher les composants qui sont appelés dans le composant UserComponent de la figure 1, au module UserModule et les enlever depuis le module racine.
- 3- Attacher le composant ProductComponent au module ProductModule et l'enlever depuis le module racine.
- 4- Attacher le composant ProviderComponent au module ProviderModule et l'enlever depuis le module racine.
- 5- Laisser les composants AppComponent, NavBarComponent, FooterComponent, HomeComponent et NotFoundComponent attachés au module racine.

- 6- Modifier les routes dans le fichier AppRoutingModuleModule comme suit :

```
const routes: Routes = [
  {path: "home", component: HomeComponent},
  {path: 'users', loadChildren: () =>
    import('./user/user.module').then(m => m.UserModule)},
  {path: 'products', loadChildren: () =>
    import('./product/product.module').then(m => m.ProductModule)},
  {path: 'providers', loadChildren: () =>
    import('./provider/provider.module').then(m => m.ProviderModule)},
  {path: "**", component: NotFoundComponent}];
```

- 7- Ajouter des routes dans le module UserRoutingModule comme suit :

```
const routes: Routes = [
  {path: "", component: UserComponent},
  {path: "user-details/:id", component: UserDetailsComponent},
];
@NgModule({
  imports: [RouterModule.forChild(routes)],
  exports: [RouterModule]
})
export class UserRoutingModule { }
```

- 8- Ajouter une route dans le module ProductRoutingModule comme suit :

```
const routes: Routes = [
  {path: "", component: ProductComponent}
];
```

- 9- Ajouter une route dans le module ProviderRoutingModule comme suit :

```
const routes: Routes = [
  {path: "", component: ProviderComponent}
];
```

Lorsque vous lancez votre projet via la commande `ng serve`, vous allez constater au niveau du terminal les modules qui sont lazy loaded

Lazy Chunk Files	Names	Raw Size
src_app_user_user_module_ts.js	user-user-module	43.36 kB
src_app_provider_provider_module_ts.js	provider-provider-module	5.47 kB
src_app_product_product_module_ts.js	product-product-module	5.44 kB