

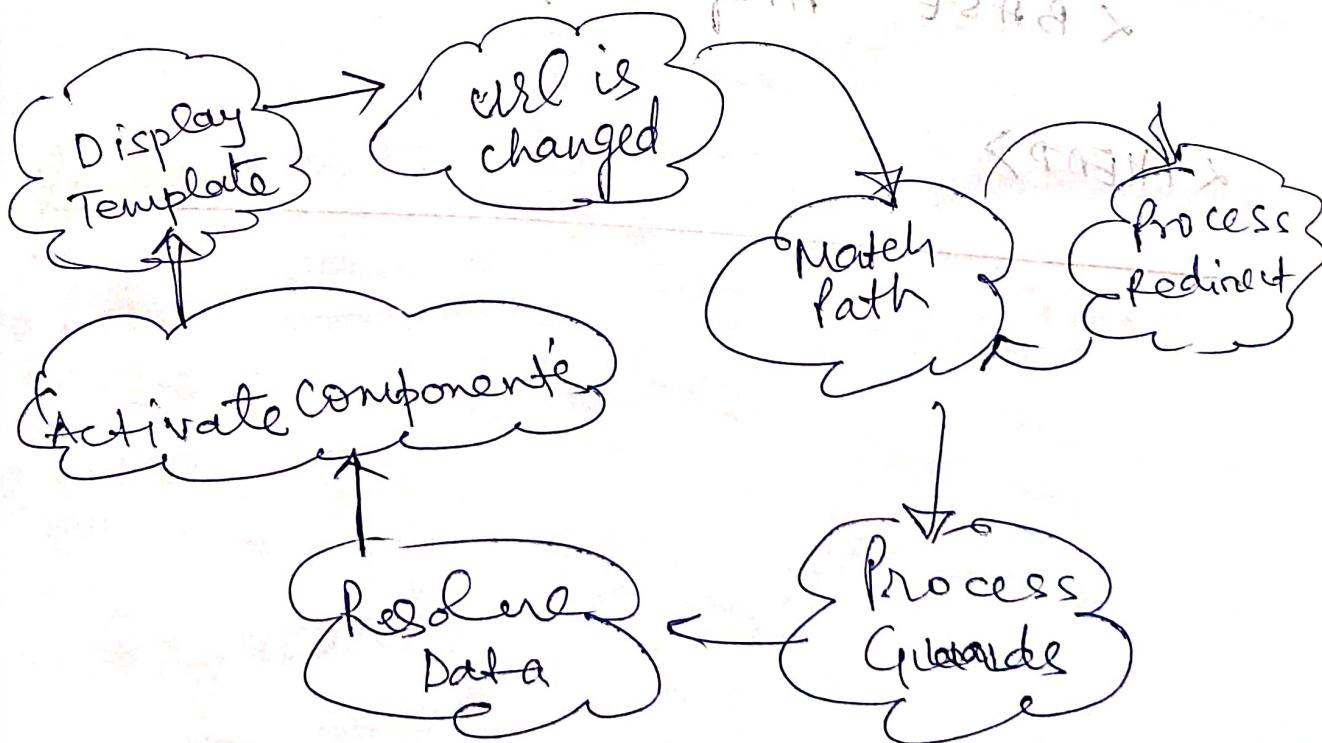
Angular Routing

- ① child and secondary routes
- ② route parameters
- ③ styling, animating and reacting
- ④ route guards and resolver.
- ⑤ lazy loading

Primary Routes

↓
child routes

↓
Secondary routes



Router - Outlet
Life cycle.

Pre-requisites

- Angular Modules
- Components
- Templates
- Services

Setting the base path

-ng build --base-href /APM

<base href = "/APM">

<HEAD>

<BASE href = "/APM">

<HEAD>

Router Module

→ Directives → ① Router Link

↳ service

② RouterLinkActive

↳ Configuration

③ Router Outlet

import { RouterModule } from '@angular/router'

from '@angular/router'

RouterModule.forRoot()

register the route service

used once for the application.

not used in child modules

RouterModule.forChild()

Does NOT register the route

service modules.

Used in feature modules.

(@NgModule({})

imports:

RouterModule.forRoot()

])

Configuring Routes

```
{ path: '/welcome', component: Welcome },  
{ path: '/', redirectTo: '/welcome' },  
{ path: '/**', components: { PageNotFound } }
```

Note: Order in routes matters a lot.
Router uses a first match wins strategy.

Activating a Route with code

```
this.router.navigate(['/home']) - standard
```

```
this.router.navigate('/home') - shortcut
```

```
this.router.navigateByURL('/welcome')
```

complete URL path

It removes the secondary routes

Secondary Routes

Can be used for → Dashboard
→ Multi-window user interface
→ Notes or comments
→ Messages.

```
<router-outlet name="popup">  
</router-outlet>
```

```
<RouterModule> for child [ ]
```

```
{ path: 'messages',  
component: MessageComponent,  
outlet: 'popup'
```

```
<a routerLink="" [outlets:
```

```
{ popup: [ 'messages' ] } >
```

```
{ ] } > message </a>
```

Activating Secondary Routes in Code

```
this.router.navigate(['$outlets:  
{popUp: [messages]33}']);
```

Clearing Secondary Routes

```
[a routerLink = "[$  
]">]  
outlet: {popUp: null 33}];
```

the outlet is telling

Route Guards

- can activate
- can activate child
- can deactivate
- can load
- resolve (data prefetch)

Guard Processing

- can deactivate
 - ↳ can load
 - ↳ can activate child
- ↳ can activate
 - ↳ resolve

Building a Guard as Service

auth.guard.ts

```
import { Injectable } from '@angular/core'  
import { CanActivate } from '@angular/router'  
{  
  canActivate(): boolean {  
    // ...  
  }  
}
```

Guarding a Route

{ path : '/id',
component : _____,

resolve : { product : ProductRes }

canActivate : [AuthGuard]

2

CLB

binds spring user/auth

workflows &
storages

service(s) to be used as provider

st. be up. st. up

dependency injection
(a set of objects that provide
services, components, and services)

3 understand: () straightforward