**Lesson34 Observable Retry Error**

**Notes:-**

**1-If we want to apply the retry sending request if the error has happen**

import { Component, OnInit } from '@angular/core';

import {EmployeeService} from '../../Service/employee.service';

import {Employee} from '../../Service/Employee';

import {TestServiceService} from '../../Service/test-service.service';

// Import rxjs retry operator

import 'rxjs/add/operator/retry';

@Component({

selector: 'app-employee-list',

templateUrl: './employee-list.component.html',

styleUrls: ['./employee-list.component.css']})

export class EmployeeListComponent implements OnInit {

MessageStatus:string;

constructor(public employeeService:EmployeeService){

this.MessageStatus = "Loading Employees .... Please Wait";}

ngOnInit() {

this.ShowEmployees();}

selectedEmployeeCountRadioButton: string = 'All';

onEmployeeCountRadioButtonChange(selectedRadioButtonValue: string): void {

this.selectedEmployeeCountRadioButton = selectedRadioButtonValue;}

GetFullName(Fname:string,Lname:string):string{

return Fname+" "+Lname;}

ShowEmployees(){

this.employeeService.GetPersons()

//If we want to apply the Retry sending request 3 times if the request has failed sending , we have to use this syntax

.retry(3)

.subscribe(data =>

this.employeeService.Employees = data as Employee[],

error => {

this.MessageStatus =

'Problem with the service. Please try again after sometime';

// Notice here we are logging the error to the browser console

console.error(error);

});}

**2-If we want to make delay time in each time the request**

// Import rxjs retry operator

import 'rxjs/add/operator/retry';

//Import rxjs retry when operator

import 'rxjs/add/operator/retrywhen';

import 'rxjs/add/operator/delay';

//we see that we apply the delay the resend request if the request has failed to unlimited times

ShowEmployees(){

this.employeeService.GetPersons()

//.retry(3)

.retryWhen((err) => err.delay(5000))

.subscribe(data =>

this.employeeService.Employees = data as Employee[],

error => {

this.MessageStatus =

'Problem with the service. Please try again after sometime';

// Notice here we are logging the error to the browser console

console.error(error);

});}

**3-If we want to mix the make fix number of retry plus detect the delay time**

// Import rxjs retry operator

import 'rxjs/add/operator/retry';

//Import rxjs and to Show the message

import 'rxjs/add/operator/retrywhen';

import 'rxjs/add/operator/delay';

import 'rxjs/add/operator/scan';

ShowEmployees(){

this.employeeService.GetPersons()

//we mix the number of retry plus detect the delay time

.retryWhen((err) => {

return err.scan((retryCount, val) => {

retryCount += 1;

if (retryCount < 6) {

this.MessageStatus = 'Retrying...Attempt #' + retryCount;

return retryCount;}

else {

throw (err);}}, 0).delay(1000)})

**And in the view we write the following code**

<table>

<thead><tr><th colspan="4">Employee Details</th></tr>

<tr><td></td>

<th style="text-align: left;">Full Name</th>

<th style="text-align: left;">Gender</th>

<th style="text-align: left;">Salary</th>

</tr></thead>

<tbody>

<tr \*ngFor="let employee of employeeService.Employees;let count = index;">

<td>{{count}}</td>

<td>{{GetFullName(employee.Fname,employee.Lname)}}</td>

<td>{{employee.Gender}}</td>

<td>{{employee.Salary | currency:'USD':true:'1.4-4'}}</td>

<td>

<a routerLink="/detail/{{employee.ID}}">Click Here</a>

</td>

</tr>

//This is the part that is used to show the no Retry Message and the error message if all the retry has failed

<tr \*ngIf="!employeeService.Employees">

<td colspan="4">

{{MessageStatus}}

</td>

</tr>

<tr \*ngIf="employeeService.Employees && employeeService.Employees.length == 0">

<td colspan="4">

No Employees to display

</td>

</tr>

</tbody>

</table>