**Lesson05 Custom Currency Component**

**Notes:-**

**1-on the currency.html we set the following code**

<form [formGroup]="currencyForm">

<div class="form-group col-md-8">

<p-dropdown [options]="currencies" formControlName="CurrencyCode" class="form-control-ng" (onChange)="changeCurrency($event)" [required]="requiredVal" filter="true">

</p-dropdown>

<span class="ng-validation-icon"></span>

</div>

<div class="form-group col-md-4">

<input type="text" formControlName="ExrateValue" class="form-control" readonly />

</div>

</form>

**2-on the currency component we set the following code**

import { Component, OnInit, Input, EventEmitter, Output, SimpleChange, SimpleChanges } from '@angular/core';

import { FormGroup, Validators, FormControl } from '@angular/forms';

import { CommonService } from '../../../shared/services/common.service';

import { Observable } from 'rxjs';

import { debug } from 'util';

import { SelectItem } from 'primeng/api';

import { Currency } from '../../../models/data-models';

@Component({

selector: 'ribms-currency',

templateUrl: './currency.component.html',

styleUrls: ['./currency.component.scss']})

export class CurrencyComponent implements OnInit {

//to make the form control required

requiredVal: Boolean;

currencyForm: FormGroup;

currencies: SelectItem[];

//we pass the Currency State that make it required

@Input() CurrencyState: Boolean;

@Input() currencyObj: Currency = new Currency();

constructor(private cs: CommonService) {}

//we create Currency Form and fill currency Drop Down and set validations

ngOnInit() {

this.createCurrencyForm();

this.fillCurrencyDropDown();

this.setValidation();}

fillCurrencyDropDown() {

this.cs.getCurrencies().subscribe(data =>

this.currencies = data,

err => console.log(err));}

setValidation() {

const currencyCode = this.currencyForm.get('CurrencyCode');

const exchangeRate = this.currencyForm.get('ExrateValue');

if (this.CurrencyState) {

currencyCode.setValidators(Validators.required);

exchangeRate.setValidators(Validators.required);

this.requiredVal = true;}

else {

currencyCode.clearValidators();

exchangeRate.clearValidators();

this.requiredVal = false;}

currencyCode.updateValueAndValidity();

exchangeRate.updateValueAndValidity();}

changeCurrency(event: any) {

const currencyCode = event.value;

if (currencyCode != "") {

this.cs.getExchangeRate(currencyCode).subscribe(

data => {

this.currencyForm.controls.ExrateValue.setValue(data);

this.currencyObj.CurrencyCode = currencyCode;

this.currencyObj.ExrateValue = data as number;});}}

editMode(CRG\_CUR\_CODE:string){

if (CRG\_CUR\_CODE != ""){

this.cs.getExchangeRate(CRG\_CUR\_CODE).subscribe(

data => {

this.currencyForm.patchValue({

CurrencyCode: CRG\_CUR\_CODE,

ExrateValue: data.toString()});

this.currencyObj.CurrencyCode = CRG\_CUR\_CODE;

this.currencyObj.ExrateValue = data as number;});}}

createCurrencyForm(){

this.currencyForm = this.cs.fb.group({

CurrencyCode: [''],

ExrateValue: ['']});}

beginAddMode(){

this.cs.resetForm(this.currencyForm);

this.createCurrencyForm();}}

**3-on the html component we set the following code**

<div class="form-group col-md-4">

<label class="control-label col-sm-5 col-xs-12">Claim Currency</label>

<ul class="input-group col-sm-7 col-xs-12">

<li>

<ribms-currency [CurrencyState]="true" [currencyObj]="clmCur" #clmChldCur></ribms-currency></li></ul></div>

<div class="form-group col-md-4">

<label class="control-label col-sm-5 col-xs-12">Claim Settling Currency</label><ul class="input-group col-sm-7 col-xs-12">

<li><ribms-currency [CurrencyState]="true" [currencyObj]="clmSetCur" #clmChldSetCur></ribms-currency></li></ul></div>

**4-on the code behind, we set the following code**

//we make the 2 reference variable that reference for the currency element

//to enable the parent to access to any method on the child compoennt

@ViewChild('clmChldCur') private claimCurrency: CurrencyComponent;

@ViewChild('clmChldSetCur') private claimSetCurrency: CurrencyComponent;

resetClaimForm() {

//we call beginAddMode() that

this.claimCurrency.beginAddMode();

this.claimSetCurrency.beginAddMode();}

setClaimFormValues() {

this.clmCur.CurrencyCode = this.claim.CrgCurCodeClaim;

this.clmSetCur.CurrencyCode = this.claim.CrgCurCodeSett;

this.claimCurrency.editMode(this.clmCur.CurrencyCode);

this.claimSetCurrency.editMode(this.clmSetCur.CurrencyCode);}

**5-on the model.ts we define the clasess**

export class Currency {

ExrateValue: number;

CurrencyCode: string;

constructor(value = 0, code = "") {

this.CurrencyCode = code;

this.ExrateValue = value;}}