

```

/*
 * A class to handle currencies
 * This class is used on the admin pages to manage currencies with
 * Ajax and JQuery.
 * It provides a list of existing currencies.
 * The user can edit and add new currencies.
 * Editing currencies happens in a jQuery UI dialog
 * box and all transactions are done with Ajax
 */
$(document).ready(function() {

    // Define edit dialog box
    $('#EditCurrencyDialogBox').dialog({modal: true ,
        autoOpen: false,
        title: 'Currency',
        width:1200
    });

    // Add events
    $('#addCurrencyButton').bind('click', function(e) {
        e.preventDefault();
        CurrencyObj.addCurrency();
    });

    // Initialize currency list
    CurrencyObj.getCurrencies();
})

var CurrencyObject = new Object();

/**
 * constructor
 *
 */
CurrencyObject = function(translationsObj )
{
    // Get translations
    this.translationsObj = translationsObj;
}

CurrencyObject.prototype =
{
    /**
     * Send Ajax request to get currencies
     *
     */
    getCurrencies:function()
    {
        var thisObj = this;
        var filtersObj = new Object();

        var jsonStr = JSON.stringify(filtersObj);

        showLoadingAjax();
        jQuery.ajax({
            url:         '/clientsadmincurrency/getallajax',
            type:         'GET',
            dataType:     'json',
            data:         {json:jsonStr},
            success: function(json) { ajaxResponse(json)}
        });
        function ajaxResponse(json)

```

```

        {
            thisObj.renderCurrencyList(json, thisObj);
            hideLoadingAjax()
        }
    },

    /**
     * Render currency from Ajax response
     */
    renderCurrencyList:function(json, thisObj)
    {
        $('#currencyTable tbody').html('');

        var rowClass = '';
        $.each(json, function(i, item) {
            var newRow = $('#templateRowCurrency').clone();
            rowClass = (rowClass == 'even') ? 'odd' : 'even';
            newRow.addClass(rowClass);
            newRow.attr('id', item.id)
            newRow.find('.currencyTableId').text(item.id);
            newRow.find('.currencyTableName').text(item.name);
            newRow.find('.currencyTableDescription').text(item.description);
            newRow.css('display' , 'tablerow')

            $('#currencyTable tbody').append(newRow);
        });

        thisObj.addEventsForCurrencyList(thisObj);
    },

    /**
     * addEventsForCurrencyList
     *
     * This is a sub function of function 'renderCurrencyList'
     * to create the button events for the buttons on the currency
     * list.
     */
    addEventsForCurrencyList:function(thisObj)
    {
        $('.currencyTableEditButton').unbind('click');
        $('.currencyTableEditButton').bind('click', function(e) {
            e.preventDefault();
            thisObj.editCurrency($(this));
        });
        // Set common event haddles in the core classes
        setCoreFormEvents();
    },

    /**
     * Delete currency
     *
     */
    deleteCurrency:function(button)
    {
        var thisObj = this;
        var _id = button.parents('tr').attr('id');
        if (confirm(this.translationsObj.confirmDelete )) {
            showLoadingAjax();
            jQuery.ajax({

```

```

        url:          '/clientsadmincurrency/deletecurrencyajax/id/' +
_id,
        type:          'GET',
        success: function() {ajaxResponse()}
    });
}

function ajaxResponse()
{
    thisObj.getCurrencies();
    hideLoadingAjax();
}
},

/**
 * Add currency, open form dialog box to add currency
 *
 */
addCurrency:function(button)
{
    var thisObj = this;
    showLoadingAjax();
    jQuery.ajax({
        url:          '/clientsadmincurrency/geteditiformajax',
        type:          'GET',
        success: function(data) {ajaxResponse(data)}
    });

    function ajaxResponse(data)
    {
        thisObj.openEditCurrencyForm(data, thisObj);
        hideLoadingAjax();
    }
},

/**
 * Edit currency
 *
 */
editCurrency:function(button)
{
    var thisObj = this;
    var _id = button.parents('tr').attr('id');

    showLoadingAjax();
    jQuery.ajax({
        url:          '/clientsadmincurrency/geteditiformajax/id/' + _id,
        type:          'GET',
        success: function(data) {ajaxResponse(data)}
    });

    function ajaxResponse(data)
    {
        CurrencyObj.openEditCurrencyForm(data, thisObj);
        hideLoadingAjax();
    }
},

```

```

/**
 * Open the form after ajax response to add or edit currency
 *
 */
openEditCurrencyForm:function(data, thisObj)
{
    $('#EditCurrencyDialogBox').dialog('open');
    $('#EditCurrencyDialogBox').html(data);
    $('#submitEditCurrencyForm').bind('click', function(e) {
        e.preventDefault();
        thisObj.saveForm();
    });
},

/**
 * Save Form
 *
 */
saveForm:function()
{
    var thisObj = this;
    showLoadingAjax();
    jQuery.ajax({
        url:         '/clientsadmincurrency/saveeditiformajax',
        type:         'POST',
        dataType:     'json',
        data:         $('#currencyForm').serialize(),
        success: function(json) {ajaxResponse(json)}
    });
    function ajaxResponse(json)
    {
        if (json.status == 'ERROR' ) {
            thisObj.openEditCurrencyForm(json.form, thisObj);
        } else {
            $('#EditCurrencyDialogBox').dialog('close');
            thisObj.getCurrencies();
        }
        hideLoadingAjax();
    }
}

} // prototype

```