```
$ = jQuery.noConflict();
$(document).ready(function() {
    $("#perfectMinDetailsToolTip").hide();
    $("#MagicSlider").css("visibility", "hidden");
    //===========================
    // show the grid for errorlog
    //===========================
    var url = "/Inventory/dashboardjson";
    // prepare the data
    var sourceTwo = {
        datatype: "json",
        datafields: [{
            name: 'TimeStamp',
            type: "date"
        }, {
            name: 'MerchantID',
            type: "string"
        }, {
            name: 'KiOuiUserId',
            type: "string"
        },  {
            name: 'MarketplaceID',
            type: "string"
        },  {
            name: 'item-name',
            type: "string"
        }, {
            name: 'Shipping',
            type: "float"
        }, {
            name: 'seller-sku',
            type: "string"
        }, {
            name: 'currencycode',
            type: "string"
        },  {
            name: 'listing-id',
            type: "string"
        }, {
            name: 'price',
            type: "number"
        },  {
            name: 'MarginGoal',
            type: "float"
        },   {
            name: 'MagicNumber',
            type: "float"
        },  {
            name: 'RepricerEnabled',
            type: "int"
        },  {
            name: 'lowestlandedprice',
            type: "string"
        },  {
            name: 'dowehavebuybox',
            type: "string"
        },  {
            name: 'TargetPrice',
            type: "number"
        },  {
            name: 'TotalEstimatedFees',
            type: "number"
        }, {
            name: 'Cost',
            type: "number"
        },  {
            name: 'priceMax',
            type: "number"
        },  {
            name: 'priceMin',
            type: "number"
        },   {
            name: 'id',
            type: "int"
        }, {
            name: 'quantity',
            type: "number"
        }, {
            name: 'fulfillment-channel',
            type: "string"
        }, {
            name: 'item-condition',
            type: "int"
        }, {
            name: 'asin1',
            type: "string"
        },  {
            name: 'category',
            type: "string"
        },   {
            name: 'salesrank',
            type: "string"
        },   {
            name: 'ActiveRuleID',
            type: "int"
        },  {
            name: 'whoSetPriceMin',
            type: "string"
        }, {
            name: 'smallimage',
            type: "string"
        },  {
            name: 'AfnTotalQuantity',
            type: "string"
        },  {
            name: 'largeimage',
            type: "string"
        }, ],
        id: 'Id',
        url: url,
        pagesize: 25,
        root: 'Rows',
        filter: function()
                {
            $("#jqxgrid").jqxGrid('updatebounddata');
            _loadDoWeHaveBuyBox();
                },

        sort: function()
                {
                        $("#jqxgrid").jqxGrid('updatebounddata', 'sort');
            _loadDoWeHaveBuyBox();
                },
        beforeprocessing: function(data)
                {
                        if (data != null)
                        {
                                sourceTwo.totalrecords = data.TotalRows;
                        }
                },
    };


    var dataAdapter = new $.jqx.dataAdapter(sourceTwo, {
        loadComplete: function(data) {
    },
        loadError: function(xhr, status, error) {
        console.log(sourceTwo);
        console.log(error);
    },
    beforeLoadComplete: function (records, originalData, c) {
        // ------------------------------------------------------------
        // if we have an empty quantity we display inbound Quantity
        // ------------------------------------------------------------
        var datainformation = $('#jqxgrid').jqxGrid('getdatainformation');
        var paginginformation = datainformation.paginginformation;
        var pagenum = paginginformation.pagenum;
        var pagesize = paginginformation.pagesize;
                    for (var i = (pagenum*pagesize); i < (pagenum*pagesize)+pagesize && i <
                    records.length-1; i++) {
            if(records[i].quantity=='') {
                records[i].quantity='('+records[i].AfnTotalQuantity+')';
            }
                    };
                    return records;
                },
    });

    var expandData = new Array();
    var cellsrenderer = function (row, column, value) {
        return '<div style="text-align: center; margin-top: 0px;">' + value + '</div>';
    }
    var columnrenderer = function (value) {
        return '<div style="text-align: center; margin-top: 4px;">' + value + '</div>';
    }


    var pricerenderer = function (row, column, value, rowdata) {
        var dataRecord = $("#jqxgrid").jqxGrid('getrowdata', row);
        if (value !='') {
            var value = parseFloat(value).toFixed(2);
            return '<div style="text-align: right; margin-top: 0px; margin-right: 12px;">' +
            dataRecord.currencycode + ' ' + value + '</div>';
        }
    }

    var checkrenderer = function (row, column, value, rowdata) {
        var dataRecord = $("#jqxgrid").jqxGrid('getrowdata', row);
        var enabled = '';
        if (value == 1) {
            enabled = '<img src="/images/check_12x12.png"  width="8px" style="padding: 8px;"/>';
        } else {
            enabled = '';
        }
        return enabled;
    }

    var buyboxrenderer = function (row, column, value, rowdata) {
        var dataRecord = $("#jqxgrid").jqxGrid('getrowdata', row);
        var have = '';
        if (dataRecord.dowehavebuybox === true) {
            have = '<img src="/images/buybox_12x12.png" style="padding: 6px;"/>';
        } else {
            have = '';
        }
        return have;
    }


    var editrenderer = function (row, column, value) {
        if (value !='') {
            var value = parseFloat(value).toFixed(2);
            return '<div style="text-align: right; margin: 3px 5px; border: solid 1px
            LightGray;">' + value + '</div>';
        } else {
                return '<div style="text-align: right; margin: 3px 5px; border: solid 1px
                LightGray;">&nbsp;</div>';
        }
    }

    // initialize jqxGrid
    $("#jqxgrid").jqxGrid({
        width: '100%',
        theme: KiouiTheme,
        source: dataAdapter,
        altrows: true,
        enabletooltips: true,
        pagesizeoptions: ['25', '50', '100',],
        showfilterrow: true,
        autoshowfiltericon: true,
        filterable: true,
        sortable: true,
        autoheight: true,
        pageable: true,
        virtualmode: true,
        selectionmode: 'singlecell',
        columnsresize: true,
        rowdetails: true,
        enablebrowserselection: true,
        showrowdetailscolumn: false,
        showsortmenuitems: false,
        rendergridrows: function(obj)
        {
            return obj.data;
        },


    rendered: function() {
    },
        ready: function() {
        _loadDoWeHaveBuyBox();
            $("#jqxgrid").on("rowclick", function(event) {
                var column = event.args.column;
                var rowindex = event.args.rowindex;
                var columnindex = event.args.columnindex;
            });
        $("#jqxgrid").on("pagechanged", function (event)
        {
        _loadDoWeHaveBuyBox();
        });

        _createjqxDropdownListRepricing();
        },
        //initrowdetails: initrowdetails,
        columns: [ {
            text: 'SKU',
            datafield: 'seller-sku',
            rendered: function (element) {
                $(element).jqxTooltip({ position: 'mouse', content: "Click Column Headers To
                Sort<br>Click Box To Search" });
            },
            width: '24%',
            menu: false,
        }, {
            text: 'Title',
            datafield: 'item-name',
            width: '20%'
        }, {
            text: '',
            datafield: 'smallimage',
            sortable: false,
            renderer: columnrenderer,
            cellsrenderer: cellsrenderer,
            enabletooltips: false,
            width: '4%',
            filterable: false
        }, {
            text: 'Condition',
            datafield: 'item-condition',
            renderer: columnrenderer,
            cellsrenderer: cellsrenderer,
            width: '6%'
        }, {
            text: 'Fulfillment Channel',
            datafield: 'fulfillment-channel',
            renderer: columnrenderer,
            cellsrenderer: cellsrenderer,
            width: '15%'
        }, {
            text: 'Price',
            datafield: 'price',
            type: 'float',
            cellsformat: 'f2',
            renderer: columnrenderer,
            cellsrenderer:  pricerenderer,
            width: '7%'
        },  {
            text: '<img src="/images/check_12x12.png"  width="8px" style="padding: 8px;"/>',
            datafield: 'RepricerEnabled',
            sortable: false,
            renderer: columnrenderer,
            cellsrenderer:  checkrenderer,
                    width: '2%',
            filterable: false,
            enabletooltips: false,
        },   {
            text: '<img src="/images/buybox_12x12.png" style="padding: 6px; float: right;"/>',
            datafield: 'dowehavebuybox',
            renderer: columnrenderer,
            //cellsrenderer:  buyboxrenderer,
            width: '2%',
            sortable: false,
            filterable: false,
            enabletooltips: false,
        }, {
            text: 'Minimum',
            datafield: 'priceMin',
            //cellsformat: 'd',
            renderer: columnrenderer,
            cellsrenderer: editrenderer,
            width: '7%'
        }, {
            text: 'Maximum',
            datafield: 'priceMax',
            //cellsformat: 'd',
            renderer: columnrenderer,
            cellsrenderer: editrenderer,
            width: '7%'
        }, {
            text: 'Quantity',
            datafield: 'quantity',
            //cellsformat: 'n',
            renderer: columnrenderer,
            cellsrenderer: cellsrenderer,
            width: '6%'
        }, ],
    });


    //
    $("#jqxgrid").on("cellclick", function(event) {
        var column = event.args.column;
        var rowindex = event.args.rowindex;
        var columnindex = event.args.columnindex;



        //when user wants to edit the record
        if (columnindex >= 0) { // columnindex 0 is first column in grid from id: 'Id' in above
        code
        // open the popup window when the user clicks price box
                editrow = rowindex;
                var offset = $("#jqxgrid").offset();
                // get the clicked row's data and initialize the input fields.
                var dataRecord = $("#jqxgrid").jqxGrid('getrowdata', editrow);



            $("#KiOuiUserId").val(dataRecord.KiOuiUserId);
            $("#MerchantID").val(dataRecord.MerchantID);
            $("#idSkuList").val(dataRecord.id);
            $("#item-name").text(dataRecord["item-name"]);
            $("#seller-sku").text(dataRecord["seller-sku"]);
            $("#asin1").html('<a target=_Blank href="http://www.amazon.com/gp/product/'+
            dataRecord.asin1+'/?tag=kimagic-20">'+dataRecord.asin1+'</a>');
            $("#salesrank").text(dataRecord.salesrank);
            $("#category").text(dataRecord.productcategory);
            $("#ActiveRuleID").val(dataRecord.ActiveRuleID);
            $("#currencycode").html(dataRecord.currencycode);
            $("#dowehavebuybox").html(dataRecord.dowehavebuybox);
            $("#RepricerEnabled").val(dataRecord.RepricerEnabled);
            $("#largeimage").html(dataRecord.largeimage);
            $("#lowestlandedprice").text(dataRecord.lowestlandedprice);//instead of
            lowestlandedprice this variable actually returns the current buy box price
            //conditionals
            if (dataRecord.price !='') {
                dataRecord.price = parseFloat(dataRecord.price).toFixed(2);
            }
            $("#price").html(dataRecord.price);

            if (dataRecord.Cost !='') {
                dataRecord.Cost = parseFloat(dataRecord.Cost).toFixed(2);
            }
            $("#Cost").val(dataRecord.Cost);

            if (dataRecord.TargetPrice !='') {
                dataRecord.TargetPrice = parseFloat(dataRecord.TargetPrice).toFixed(2);
            }
            $("#TargetPrice").val(dataRecord.TargetPrice);

            if (dataRecord.priceMin !='') {
                dataRecord.priceMin = parseFloat(dataRecord.priceMin).toFixed(2);
                $("#priceMin").val(dataRecord.priceMin);
            } else {
                $("#priceMin").val(dataRecord.priceMin);
            }

            if (dataRecord.priceMax !='') {
                dataRecord.priceMax = parseFloat(dataRecord.priceMax).toFixed(2);
                $("#priceMax").val(dataRecord.priceMax);
            } else {
                $("#priceMax").val(dataRecord.priceMax);
                    }

            //api loads
            $("#salesrank").html('<img src="/images/loader.gif" style="width: 15px;"/></span>');
            $("#category").html('<img src="/images/loader.gif" style="width: 15px;"/></span>');
            $("#lowestlandedprice").html('<img src="/images/loader.gif" style="width:
            15px;"/></span>');
            $("#totalestimatedfees").html('<img src="/images/loader.gif" style="width:
            15px;"/></span>');




        // shows the popup window.
        showPopUpEditWindow(dataRecord);


        // Create jqxTabs.
        $('#jqxTabs').jqxTabs({
            width: 730,
            height: 340,
            });
        $('#jqxTabs').jqxTabs({ selectedItem: 0 });


        $('#category').load('/ProductAPI/SalesRankFromASIN/'+dataRecord.asin1+'/'+dataRecord.
        MarketplaceID+' #category');
        $('#salesrank').load('/ProductAPI/SalesRankFromASIN/'+dataRecord.asin1+'/'+dataRecord.
        MarketplaceID+' #salesrank');
        $('#lowestlandedprice').load('/ProductAPI/CompetitivePricingForASIN/'+dataRecord.asin1+
        '/'+dataRecord.MarketplaceID+' #lowest');
        //instead of lowestlandedprice this variable actually returns the current buy box price

        $("#jqxcheckbox").jqxCheckBox();

        if (dataRecord.RepricerEnabled === 1) {
            $("#jqxcheckbox").jqxCheckBox({ checked: true });
            $("#RepricerEnabled").html('Repricer Enabled');
        } else {
            $("#jqxcheckbox").jqxCheckBox({ checked: false });
            $("#RepricerEnabled").html('Repricer Disabled');
        }


        _loadDoWeHaveBuyBox();
        var dowe = ((dataRecord.dowehavebuybox).toString()).charAt(1);
        if (dowe == 'i') {
            $("#dowehavebuybox").html('<img src="/images/buybox_12x12.png" style="margin: 3px;
            float: right; margin-right: -21px;"/>');
        } else {
            $("#dowehavebuybox").html('');
        }


        // ----------------------------------
        // prepopulate ActiveRuleID
        // ----------------------------------
        if(dataRecord.ActiveRuleID == 99 || dataRecord.ActiveRuleID === null) {
            indexDropDownOne = $("#jqxDropdownListRepricing").jqxDropDownList('getItemByValue',
            recordDefaults.DefaultRepricerRuleID).index;
        } else {
            indexDropDownOne = $("#jqxDropdownListRepricing").jqxDropDownList('getItemByValue',
            dataRecord.ActiveRuleID).index;
        }
        $("#jqxDropdownListRepricing").jqxDropDownList({selectedIndex: indexDropDownOne });


        $("#jqxcheckbox").bind('change', function (event) {
                var checked = event.args.checked;
                if (checked === true) {
                    $("#RepricerEnabled").val(1)
                    $("#RepricerEnabled").html('Repricer Enabled');
                } else {
                    $("#RepricerEnabled").val(0)
                    $("#RepricerEnabled").html('Repricer Disabled');
                }
            });


        // ----------------------------------------------------
        // we wait 300's of a second and we clear selection
        // the problem is that the program is too fast to
        // open and has no time to clear the selection
        // the mouse is still moving for a micro second when
        // the popup opens so we wait 300 mili seconds and
        // we clear the selection
        // ----------------------------------------------------
        setTimeout(function () {
                if (window.getSelection) {
                  if (window.getSelection().empty) {  // Chrome
                    window.getSelection().empty();
                  } else if (window.getSelection().removeAllRanges) {  // Firefox
                    window.getSelection().removeAllRanges();
                  }
                } else if (document.selection) {  // IE?
                  document.selection.empty();
                }
        }, 300);

        };
    });

    //validation rules for user entry of minmax prices
    $('#edittemplateForm').jqxValidator({
        rules: [
        { input: '#Cost',
            message: 'Cost must be blank OR a number greater than $0.01.',
            rule: function (input, commit){
                var value = $("#Cost").val();
                if (value == "") {
                    return true;
                } else if (value > 0 && (!isNaN(value))) {
                    return true;
                }
                return false;
            }
        },{ input: '#MarginGoal',
            message: 'Margin Goal must be blank OR a number',
            rule: function (input, commit){
                var value = $("#MarginGoal").val();
                if (value == "") {
                    return true;
                } else if (value >!isNaN(value)) {
                    return true;
                }
                return false;
            }
        },{ input: '#TargetPrice',
            message: 'Target price must be blank OR a number greater than $0.01.',
            rule: function (input, commit){
                var value = $("#TargetPrice").val();
                if (value == "") {
                    return true;
                } else if (value > 0 && (!isNaN(value))) {
                    return true;
                }
                return false;
            }
        },{ input: '#priceMin',
            message: 'Minimum price must be blank OR a number greater than $0.01.',
            rule: function (input, commit){
                var value = $("#priceMin").val();
                if (value == "") {
                    return true;
                } else if (value > 0 && (!isNaN(value))) {
                    return true;
                }
                return false;
            }
        },
        { input: '#priceMax',
            message: 'Maximum price must be blank OR a number greater than $0.01.',
            rule: function (input, commit){
                var value = $("#priceMax").val();
                if (value == "") {
                    return true;
                } else if (value > 0 && (!isNaN(value))) {
                    return true;
                }
                return false;
            }
        }
    ]});


    //*********************************************************************************************
    ***
    // below is ajax code to update table - ADD row
    
    //*********************************************************************************************
    ***

    // updates the edited row when the user clicks the 'Save' button.
    $("#save").click(function(event) {
            var validate = false;
            var validate = $('#edittemplateForm').jqxValidator('validate');//invoke validation
            rules
            if (validate === true) {
                var item = $("#jqxDropdownListRepricing").jqxDropDownList('getSelectedItem');
                var row = {
                    RepricerRuleId: item.value,
                    Sku: $("#seller-sku").text(),
                    KiOuiUserId: $("#KiOuiUserId").val(),
                    MerchantID: $("#MerchantID").val(),
                    asin1: $("#asin1").text(),
                    RepricerEnabled: $("#RepricerEnabled").val(),
                    MagicNumber: $("#jqxSliderEventPopUp").val(),
                    MarginGoal: $("#MarginGoal").val(),
                    Cost: $("#Cost").val(),
                    priceMin: $("#priceMin").val(),
                    TargetPrice: $("#TargetPrice").val(),
                    priceMax: $("#priceMax").val(),
                };

                // ajax call to UPDATE the row on edit
                $.ajax({
                    url: "/inventory/updateamazonskuminmaxlistajax", // we should update the
                    name to something more generic since we update more than min max jc
                    type: "POST",
                    cache: false,
                    async: false,
                    data: row,
                    success: function(response) {
                        if (response == "success") {
                            $('#PopUpEditWindow').jqxWindow('close');
                            closeAllToolTips();
                            $('#jqxgrid').jqxGrid('updatebounddata');
                            _loadDoWeHaveBuyBox();
                        }
                    }
                });
            }
    });


    //***************************above is ajax code to update
    table******************************************************************************



    //====================================================================
    //  window open in inventory/dashboard view
    //====================================================================
    var basicDemo = (function() {
        //Adding event listeners
        function _addEventListeners() {
            $('#addrowbutton1').click(function() {
                $('#PopUpEditWindow').jqxWindow('open'); // to open the window onclick of open
                button
            });
            $('#hideWindowButton').mousedown(function() {
                closeAllToolTips();
                $('#PopUpEditWindow').jqxWindow('close'); // to close the add emailtemplate
                window onclick of cancel button
            });
            $('#cancel').mousedown(function() {
                closeAllToolTips();
                $('#PopUpEditWindow').jqxWindow('close'); // to close the edit emailtemplate
                window onclick of cancel button
            });
        };

        //Creating all page elements which are jqxWidgets
        function _createElements() {
            $('#addrowbutton1').jqxButton({
                width: '70px',
                theme: KiouiTheme
            });
        };



        function _loadDefaultValues() {
                 var url = "/Defaults/dashboardjson";
                    // prepare the data
                    var sourceDefaults = {
                        datatype: "json",
                        datafields: [{
                            name: 'MarginGoal',
                            type: "string"
                        },   {
                            name: 'MagicNumber',
                            type: "float"
                        },   {
                            name: 'DefaultRepricerRuleID',
                            type: "int"
                        }],
                        id: 'Id',
                        url: url,
                        async: true
                };
                var dataAdapterDefaults = new $.jqx.dataAdapter(sourceDefaults, {
                        loadComplete: function () {
                                // get data records.
                                var recordsDefaults = dataAdapterDefaults.records;
                                recordDefaults = recordsDefaults[0];
                        }
                }
                );
                dataAdapterDefaults.dataBind();
        };

        // create the slider
        function _createSlider() {
            $("#jqxSliderEventPopUp").jqxSlider({
                            theme: 'summer',
                            width: 295,
                            mode: 'default',
                            min: 0,
                            max: 1,
                            ticksFrequency: 20,
                            step: .01,
                            showButtons: false,
                    });
        }

        //Creating the pop up edit window
        function _createWindow() {
            var offset = $("#jqxgrid").offset();

            $('#PopUpEditWindow').jqxWindow({
                maxHeight: 700,
                maxWidth: 1000,
                minHeight: 500,
                minWidth: 525,
                height: 575,
                width: 755,
                isModal: true,
                modalOpacity: 0.5,
                autoOpen: false,
                showCollapseButton: false,
                initContent: function() {
                    $('#PopUpEditWindow').jqxWindow('focus');

                },
            });
            $('#PopUpEditWindow').on('close', function (event) {
                closeAllToolTips();
            });
        };


        return {
            config: {
                dragArea: null
            },
            init: function() {
                //_createElements();
                _addEventListeners();
                _createWindow();
                _loadDefaultValues();
                _createSlider();
            }
        };



    }());
    function _loadDoWeHaveBuyBox(){
        function _doThatAjax(Row) {
                                    $.ajax({
                                            url: "/inventory/dowehavethebuyboxajax",
                                            type: "POST",
                                            cache: false,
                                            async: true,
                                            data: Row,
                                            success: function(response) {
                        Done=1;
                                                            var resp = $.parseJSON(response);
                                                            if(resp.DoWeHaveTheBuyBox==1) {
                                                                    $("#jqxgrid").jqxGrid(
                                                                    'setcellvaluebyid', resp
                                                                    .uid, 'dowehavebuybox',
                                                                    "<img
                                                                    src=\"/images/buybox_12x1
                                                                    2.png\"
                                                                    style=\"padding: 6px;
                                                                    float: right;\"/>");
                                                            } else {
                                                                    $("#jqxgrid").jqxGrid(
                                                                    'setcellvaluebyid', resp
                                                                    .uid, 'dowehavebuybox',
                                                                    '');
                                                            }
                                            },
                                    });
        }
        function doStuff() {
            var something = $('#jqxgrid').jqxGrid('getcelltextbyid', '0', "dowehavebuybox");
            if(something==null) {//we want it to match
                setTimeout(doStuff, 1000);//wait 1 sec then recheck
                return;
            }
            dataRowsForAjax = $('#jqxgrid').jqxGrid('getrows');
            for (i = 0; i < dataRowsForAjax.length; i++) {
                expandData.push(false);
                _doThatAjax(dataRowsForAjax[i]);
            }
        }
        doStuff();
    }


    function _createjqxDropdownListRepricing() {
                var url = "/Inventory/dropdownjson";
                var sourceOne = {
                        datatype: "json",
                        datafields: [{
                                name: 'id'
                        },{
                                name: 'ShortName'
                        },
                    ],
                    url: url,
                    async: false,
                };
                var dataAdapterOne = new $.jqx.dataAdapter(sourceOne);
                $("#jqxDropdownListRepricing").jqxDropDownList({ source: dataAdapterOne,
                displayMember: "ShortName", valueMember: "id", width: '200px', height: '20px' });
                $('#jqxDropdownListRepricing').bind('select', function (event) {
                    var args = event.args;
                    var item = $('#jqxDropdownListRepricing').jqxDropDownList('getItem', args.
                    index);
                    // we show or hide the slider
                    if(item.value=='7') {
                        $("#MagicSlider").css("visibility", "visible");
                    } else {
                        $("#MagicSlider").css("visibility", "hidden");
                    }
                });
    };


        //
        -----------------------------------------------------------------------------------------
        -----------
        // We have tooltips that do not auto close so we want to make sure they close on close
        of the popup
        //
        -----------------------------------------------------------------------------------------
        -----------
        function closeAllToolTips() {
            $("#totalEstimatedFeesToolTip").jqxTooltip("close");
            $("#perfectMinDetailsToolTip").jqxTooltip("close");
        }


        //
        -----------------------------------------------------------------------------------------
        -----------
        // We calculate perfect min price
        //
        -----------------------------------------------------------------------------------------
        -----------
        function calculatePerfectMinPrice() {
        var togglePerfectMin = false;
                Cost = parseFloat($("#Cost").val());
        MarginGoal = parseFloat($("#MarginGoal").val());
        if(!isNaN(Cost) && !isNaN(MarginGoal)) {
            if(FulfillmentChannel.indexOf("Amazon") > -1) {
                priceMin = $("#priceMin").val();
                if(priceMin!='') {
                    priceMin = parseFloat(priceMin);
                    priceMin = parseFloat(priceMin);
                    ReferalFeePriceMin = Math.round((priceMin * ReferalFeePercentage / 100) *
                    100) / 100;
                    if (ReferalFeePriceMin < MinimumReferalFee) {
                                                ReferalFeePriceMin = MinimumReferalFee;
                                        }
                    ReferalFeePriceMinTotal = ReferalFeePriceMin + VariableClosingFees;
                    MarginGoalCalculated = Math.round((MarginGoal * priceMin /100) * 100) / 100;
                    var TotalPriceMin =   ReferalFeePriceMinTotal +
                                                        Cost +
                                                        MarginGoalCalculated +
                                                        TotalFbaFees
                                                        ;
                                        perfectMinDetails = '<table>';
                                        perfectMinDetails += '<TR><TD align=left colspan=2
                                        style="text-transform: uppercase;"><strong>'+
                                        FulfillmentChannel+'</strong></TD></TR>';
                                        perfectMinDetails += '<TR><TD align=left>Cost:</TD><TD>'
                                        +Cost.toFixed(2)+'</TD></TR>';
                                        perfectMinDetails += '<TR><TD align=left>Margin
                                        Goal:</TD><TD>'+MarginGoalCalculated.toFixed(2)+
                                        '</TD></TR>';
                                        perfectMinDetails += '<TR><TD align=left>Referal Fee
                                        Total:</TD><TD>'+ReferalFeePriceMinTotal.toFixed(2)+
                                        '</TD></TR>';
                                        perfectMinDetails += '<TR><TD align=left>FBA
                                        Fees:</TD><TD>'+TotalFbaFees.toFixed(2)+'</TD></TR>';
                                        perfectMinDetails += '<TR><TD align=left
                                        style="border-top: 1px solid black;">Total:</TD><TD
                                        style="border-top: 1px solid black;">'+TotalPriceMin.
                                        toFixed(2)+'</TD></TR>';
                                        perfectMinDetails += '</table>';
                } else {
                                    perfectMin = Math.round( (Cost + TotalFbaFees +
                                    VariableClosingFees ) / (1-(ReferalFeePercentage/100)-(
                                    MarginGoal/100)) *100) / 100;
                    ReferalFeePerfectMin = Math.round((perfectMin * ReferalFeePercentage / 100)
                    * 100) / 100;
                    if (ReferalFeePerfectMin < MinimumReferalFee) {
                        ReferalFeePerfectMin = MinimumReferalFee;
                        perfectMin = Math.round( (Cost + TotalFbaFees + ReferalFeePerfectMin +
                        VariableClosingFees ) / (1-(MarginGoal/100)) *100) / 100;
                    }
                    MarginGoalCalculated = Math.round((MarginGoal * perfectMin /100) * 100) /
                    100;
                    ReferalFeePerfectMinTotal = ReferalFeePerfectMin + VariableClosingFees;
                                    var TotalPerfectMin =   ReferalFeePerfectMinTotal +
                            Cost +
                            MarginGoalCalculated +
                            TotalFbaFees
                                                        ;

                                    $("#priceMin").jqxInput({placeHolder: perfectMin});
                                    perfectMinDetails = '<table>';
                                    perfectMinDetails += '<TR><TD align=left colspan=2 
                                    style="text-transform: uppercase;"><strong>'+
                                    FulfillmentChannel+'</strong></TD></TR>';
                                    perfectMinDetails += '<TR><TD align=left>Cost:</TD><TD>'+
                                    Cost.toFixed(2)+'</TD></TR>';
                                    perfectMinDetails += '<TR><TD align=left>Margin 
                                    Goal:</TD><TD>'+MarginGoalCalculated.toFixed(2)+'</TD></TR>';
                                    perfectMinDetails += '<TR><TD align=left>Referal Fee 
                                    Total:</TD><TD>'+ReferalFeePerfectMinTotal.toFixed(2)+
                                    '</TD></TR>';
                                    perfectMinDetails += '<TR><TD align=left>FBA Fees:</TD><TD>'
                                    +TotalFbaFees.toFixed(2)+'</TD></TR>';
                                    perfectMinDetails += '<TR><TD align=left style="border-top: 
                                    1px solid black;">Total:</TD><TD style="border-top: 1px 
                                    solid black;">'+TotalPerfectMin.toFixed(2)+'</TD></TR>';
                                    perfectMinDetails += '</table>';
                }
                        } else {

            }
            $("#perfectMinDetailsToolTip").jqxTooltip({ position: 'right', content: 
            perfectMinDetails, autoHide: false, trigger: "none", closeOnClick: false});
            $("#perfectMinDetailsToolTip").show();
        } else {
            $("#perfectMinDetailsToolTip").jqxTooltip("close");
            $("#priceMin").jqxInput({placeHolder: ''});
                        togglePerfectMin = false;
            perfectMinDetails = '';
            $("#perfectMinDetailsToolTip").hide();
            $("#perfectMinDetailsToolTip").jqxTooltip('destroy');
                }
                $("#perfectMinDetailsToolTip").click(function () {
                    if (togglePerfectMin == false) {
                            $("#perfectMinDetailsToolTip").jqxTooltip("open");
                                togglePerfectMin = true;
                        } else {
                            $("#perfectMinDetailsToolTip").jqxTooltip("close");
                                togglePerfectMin = false;
                        }
                });
    }

    function showPopUpEditWindow(dataRecord) {
        // --------------------------------
        //GET WINDOW HEIGHT AND WIDTH
        // --------------------------------
        var winHeight = $(window).height();
        var winWidth = $(window).width();
        // --------------------------------
        //KEEP CENTERED
        // --------------------------------
        var posX = (winWidth/2) - ($('#PopUpEditWindow').width()/2) + $(window).scrollLeft();
        var posY = (winHeight/2) - ($('#PopUpEditWindow').height()/2) + $(window).scrollTop();
        $('#PopUpEditWindow').jqxWindow({position: {x: posX, y: posY}});
        // --------------------------------
        //KEEP CENTERED WHEN SCROLLING
        // --------------------------------
        $(window).scroll(function() {
            winHeight = $(window).height();
            winWidth = $(window).width();
            posX = (winWidth/2) - ($('#PopUpEditWindow').width()/2) + $(window).scrollLeft();
            posY = (winHeight/2) - ($('#PopUpEditWindow').height()/2) + $(window).scrollTop();
            $('#PopUpEditWindow').jqxWindow({position: {x: posX, y: posY}});
        });
        // ----------------------------------
        //KEEP CENTERED EVEN WHEN RESIZING
        // ----------------------------------
        $(window).resize(function() {
            winHeight = $(window).height();
            winWidth = $(window).width();
            posX = (winWidth/2) - ($('#PopUpEditWindow').width()/2) + $(window).scrollLeft();
            posY = (winHeight/2) - ($('#PopUpEditWindow').height()/2) + $(window).scrollTop();
            $('#PopUpEditWindow').jqxWindow({position: {x: posX, y: posY}});
        });

        // ----------------------------------------
        // We ajax request the fees to get json
        // ----------------------------------------
        $.ajax({
                url: "/ProductAPI/AsinFees/"+dataRecord.asin1+'/'+dataRecord.MarketplaceID,
                    type: "GET",
                    cache: false,
                    async: true,
                    success: function(response) {
                if(response!='false') {
                    dataFees = jQuery.parseJSON(response);
                    FulfillmentChannel = dataRecord['fulfillment-channel'];
                    if(FulfillmentChannel.indexOf("Amazon") > -1) {
                        LandedPrice = parseFloat(dataRecord.price) + parseFloat(dataRecord.
                        Shipping);
                        ReferalFee = (LandedPrice * parseFloat(dataFees.ReferalFeePercentage) /
                        100);
                        ReferalFeePercentage = parseFloat(dataFees.ReferalFeePercentage);
                        VariableClosingFees = parseFloat(dataFees.VariableClosingFees);
                        InboundFbaShippingCost = parseFloat(dataFees.InboundFbaShippingCost);
                        OrderHandling= parseFloat(dataFees.OrderHandling);
                        PickAndPack = parseFloat(dataFees.PickAndPack);
                        ThirtyDayStorageFee = parseFloat(dataFees.ThirtyDayStorageFee);
                        WeightHandling = parseFloat(dataFees.WeightHandling);
                        MinimumReferalFee = parseFloat(dataFees.MinimumReferalFee);
                        TotalFbaFees =          InboundFbaShippingCost +
                                                OrderHandling +
                                                PickAndPack +
                                                ThirtyDayStorageFee +
                                                WeightHandling
```

```
                                                                        ;
                        if (ReferalFee < MinimumReferalFee) ReferalFee = MinimumReferalFee;
                        var TotalFeesCurentPrice =  ReferalFee +
                                        VariableClosingFees +
                                        InboundFbaShippingCost +
                                        OrderHandling +
                                        PickAndPack  +
                                        ThirtyDayStorageFee +
                                        WeightHandling
                                        ;
                        TotalFeesCurentPrice = Math.round(TotalFeesCurentPrice*100) /100;
                        totalEstimatedFeesToolTipContent = '<table>';
                        totalEstimatedFeesToolTipContent += '<TR><TD align=left colspan=2
                        style="text-transform: uppercase;"><strong>'+FulfillmentChannel+
                        '</strong></TD></TR>';
                        totalEstimatedFeesToolTipContent += '<TR><TD align=left>Referal
                        Fee:</TD><TD>'+ReferalFee.toFixed(2)+'</TD></TR>';
                        totalEstimatedFeesToolTipContent += '<TR><TD align=left>Variable
                        Closing Cost:</TD><TD>'+VariableClosingFees.toFixed(2)+'</TD></TR>';
                        totalEstimatedFeesToolTipContent += '<TR><TD align=left>Inbound
                        Shipping:</TD><TD>'+InboundFbaShippingCost.toFixed(2)+'</TD></TR>';
                        totalEstimatedFeesToolTipContent += '<TR><TD align=left>Order
                        Handling:</TD><TD>'+OrderHandling.toFixed(2)+'</TD></TR>';
                        totalEstimatedFeesToolTipContent += '<TR><TD align=left>Pick And
                        Pack:</TD><TD>'+PickAndPack.toFixed(2)+'</TD></TR>';
                        totalEstimatedFeesToolTipContent += '<TR><TD align=left>Weight
                        Handling:</TD><TD>'+WeightHandling.toFixed(2)+'</TD></TR>';
                        totalEstimatedFeesToolTipContent += '<TR><TD align=left>Thirty Day
                        Storage:</TD><TD>'+ThirtyDayStorageFee.toFixed(2)+'</TD></TR>';
                        totalEstimatedFeesToolTipContent += '<TR><TD align=left
                        style="border-top: 1px solid black;">Total Fees:</TD><TD
                        style="border-top: 1px solid black;">'+TotalFeesCurentPrice.toFixed(2)+
                        '</TD></TR>';
                        totalEstimatedFeesToolTipContent += '</table>';
                        calculatePerfectMinPrice();
                    } else {
                        LandedPrice = parseFloat(dataRecord.price) + parseFloat(dataRecord.
                        Shipping);
                        ReferalFee = (LandedPrice * parseFloat(dataFees.ReferalFeePercentage) /
                        100);
                        if (ReferalFee < parseFloat(dataFees.MinimumReferalFee)) ReferalFee =
                        parseFloat(dataFees.MinimumReferalFee);
                        ReferalFeePercentage = parseFloat(dataFees.ReferalFeePercentage);
                                            VariableClosingFees = parseFloat(dataFees.
                                            VariableClosingFees);
                                            TotalFbaFees = 0;
                                            TotalFeesCurentPrice =      ReferalFee +
                                                                            VariableClosingFees
                                                                            ;
                                            TotalFeesCurentPrice = Math.round(
                                            TotalFeesCurentPrice*100) /100;
                                            totalEstimatedFeesToolTipContent = '<table>';
                                            totalEstimatedFeesToolTipContent += '<TR><TD
```

```
                                                align=left colspan=2  style="text-transform:
                                                uppercase;"><strong>Merchant
                                                Fulfilled</strong></TD></TR>';
                                                totalEstimatedFeesToolTipContent += '<TR><TD
                                                align=left>Referal Fee:</TD><TD>'+ReferalFee.toFixed
                                                (2)+'</TD></TR>';
                                                totalEstimatedFeesToolTipContent += '<TR><TD
                                                align=left>Variable Closing Cost:</TD><TD>'+
                                                VariableClosingFees.toFixed(2)+'</TD></TR>';
                                                totalEstimatedFeesToolTipContent += '<TR><TD
                                                align=left style="border-top: 1px solid
                                                black;">Total Fees:</TD><TD style="border-top: 1px
                                                solid black;">'+TotalFeesCurentPrice.toFixed(2)+
                                                '</TD></TR>';
                                                totalEstimatedFeesToolTipContent += '</table>';
                        calculatePerfectMinPrice();
                    }
                    $("#totalestimatedfees").html(parseFloat(TotalFeesCurentPrice).toFixed(2) +
                    "&nbsp;&nbsp;&nbsp;<img id='totalEstimatedFeesToolTip' title='details'
                    src='/images/ViewDetails.png' height=16 border=0 style='vertical-align:
                    text-bottom;'>");
                    var toggle = false;
                                        $("#totalEstimatedFeesToolTip").jqxTooltip({ position:
                                        'right', content: totalEstimatedFeesToolTipContent, autoHide
                                        : false, trigger: "none", closeOnClick: false});
                                        $("#totalEstimatedFeesToolTip").click(function () {
                                        if (toggle == false) {
                                            $("#totalEstimatedFeesToolTip").jqxTooltip("open");
                                                    toggle = true;
                                                } else {
                                                  $("#totalEstimatedFeesToolTip").jqxTooltip("close"
                                                  );
                                                    toggle = false;
                                                }
                                        });
                } else {
                    $("#totalestimatedfees").html('NA');
                }
                    }
            });

            Sku = dataRecord['seller-sku'];
            MerchantID = dataRecord.MerchantID;
            MarketplaceID = dataRecord.MarketplaceID;


            $('#jqxTabs').on('tabclick', function (event) {
                 var tabclicked = event.args.item;
                 if (tabclicked = 1) {
                     _loadSkuHistory();
                 }
             });
            function _loadSkuHistory() {

                                var url = "/inventory/amazonskuhistoryjson/"+Sku+"/"+MerchantID+
                                "/"+MarketplaceID;
                                console.log(url);

                                // prepare the data
                                var sourceHistory =
                                {
                                    datatype: "json",
                                    datafields: [
                                        { name: 'TimeStamp',map: "TimeStamp",type: "date"},

                                        { name: 'Comment',map: "Comment",type: "comment"},
                                    ],
                                    id: 'id',
                                    url: url,
                                    pagesize: 30,
                                };
                                var dataAdapterHistory = new $.jqx.dataAdapter(sourceHistory, {
                                        loadComplete: function (data) { },
                                        loadError: function (xhr, status, error) { },
                                        beforeLoadComplete: function (records, originalData, c) {
                                            for (var i = 0; i < records.length; i++) {
                                                if(originalData[i].TimeStamp=="0"){
                                                      records[i].TimeStamp =""
                                                }else{
                                                      records[i].TimeStamp = new Date(
                                                      originalData[i].TimeStamp * 1000)
                                                }
                                            };
                                            return records;
                                        },
                                    });

                                // initialize jqxGrid
                                $("#jqxgridHistory").jqxGrid(
                                {
                                    width:'100%',
                                    theme: KiouiTheme,
                                    source: dataAdapterHistory,
                                    pageable: true,
                                    autoheight: true,
                                    sortable: true,
                                    altrows: true,
                                    enabletooltips: true,
                                    pagesizeoptions: ['30', '90', '200'],
                                    showfilterrow: true,
                                    filterable: true,
                                    selectionmode: 'singlerow',
                                    enablebrowserselection: true,
                                    selectionmode: 'singlecell',
                                    columns: [
                                      { text: 'Date (PST)', datafield: 'TimeStamp', type: 'date'
                                      , cellsformat: 'M/d/yy h:mm tt', width:'25%'},
                                      { text: 'Event',  datafield: 'Comment', width:'75%'},
                                     ],

                                });
                                // to remove the filter result
                                $('#clearfilteringbuttonHistory').jqxButton({ height: 25,theme:
                                KiouiTheme});
                                $('#clearfilteringbuttonHistory').click(function () {
                                    $("#jqxgridHistory").jqxGrid('clearfilters');
                                });
            };

            // ----------------------------------
            // prepopulate magic number slider
            // ----------------------------------
                if(dataRecord.MagicNumber == 'undefined' || dataRecord.MagicNumber === null) {
                    var valueSlider = recordDefaults.MagicNumber;
                    $('#jqxSliderEventPopUp').jqxSlider('setValue', valueSlider);
                } else {
                    var valueSlider = dataRecord.MagicNumber;
                    $('#jqxSliderEventPopUp').jqxSlider('setValue', valueSlider);
                }

            // ----------------------------------
            // prepopulate margin goal
            // ----------------------------------
            if(dataRecord.MarginGoal == 'undefined' || dataRecord.MarginGoal === null) {
                $('#MarginGoal').val(recordDefaults.MarginGoal);
            } else {
                $('#MarginGoal').val(dataRecord.MarginGoal);
            }

            // ----------------------------------
            // Events for fields recalculation
            // ----------------------------------
            $('#Cost').on('change', function () {
                calculatePerfectMinPrice();
            });
            $('#MarginGoal').on('change', function () {
                            calculatePerfectMinPrice();
                    });
            $('#priceMin').on('change', function () {
                            calculatePerfectMinPrice();
                    });

            // ----------------------------------
            //OPEN WINDOW
            // -----------------------------------
            $('#PopUpEditWindow').jqxWindow('open');
```

```
    }

    $(document).ready(function() {
        basicDemo.init();
    });

});
```