

CustomerOrderWebForm App Notes

The CustomerOrderWebForm project is created using the ASP.NET WebForm template to access customer-order data in a SQL Server database.

There are three web application within this project:

- CustomerOrderORM.aspx
- CustomerOrderRaw.aspx
- CustomerOrderXml.html

Note: All the above three web application have identical function.

CustomerOrderORM.aspx

The CustomerOrderORM.aspx web application is using the WebForm template and a simple ORM (Object Relational Mapping) library to access the back-end SQL database. For entry level developer or developer with limited knowledge about ORM, the codes may be somewhat abstract and can be difficult to read and understand.

CustomerOrderRaw.aspx

The CustomerOrderORM.aspx web application is using the WebForm template and raw C# codes to access the back-end SQL database (Many refer to this method as ADO.NET).

CustomerOrderXml.html

The CustomerOrderXml.html web application is simply an HTML page with JavaScript and CSS (no framework in use) to access the back-end SQL database via a web service.

The **CustomerOrderWS.asmx** is the web service created to access back-end SQL database and provide the data in XML format.

A working app is deployed to Azure and accessible via the following URL:

<https://jsmeetup2.azurewebsites.net/CustomerOrderORM.aspx>

Customer Orders ORM

CustomerOrder CustomerOrder(HTML) Excel Export

Hide Details Display Subtotal From 01/01/2019 To 12/31/2019 Go

| OrderID | CustomerID | CompanyName | Country | SalesRep | OrderDate | Shipper | Freight | OrderTotal |
|--------------------------|------------|----------------------------|-----------|------------------|-----------|------------------|-----------|------------|
| 10409 | OCEAN | Oceano Atlántico Ltda. | Argentina | Janet Leverling | 01/09/19 | Speedy Express | 29.83 | 319.20 |
| 10440 | RANCH | Rancho grande | Argentina | Margaret Peacock | 02/17/19 | United Package | 38.82 | 443.40 |
| 10521 | CACTU | Cactus Comidas para llevar | Argentina | Laura Callahan | 04/29/19 | United Package | 17.22 | 225.50 |
| 10531 | OCEAN | Oceano Atlántico Ltda. | Argentina | Robert King | 05/08/19 | Speedy Express | 8.12 | 110.00 |
| 10716 | RANCH | Rancho grande | Argentina | Margaret Peacock | 10/24/19 | United Package | 22.57 | 706.00 |
| 10782 | CACTU | Cactus Comidas para llevar | Argentina | Anne Dodsworth | 12/17/19 | Federal Shipping | 1.10 | 12.50 |
| Subtotal (Argentina) : | | | | | | | 117.66 | 1,816.60 |
| 10795 | ERNSH | Ernst Handel | Austria | Laura Callahan | 12/24/19 | United Package | 126.86 | 2,489.25 |
| 10764 | ERNSH | Ernst Handel | Austria | Michael Suyama | 12/03/19 | Federal Shipping | 145.45 | 2,540.00 |
| 10771 | ERNSH | Ernst Handel | Austria | Anne Dodsworth | 12/10/19 | United Package | 11.19 | 344.00 |
| 10773 | ERNSH | Ernst Handel | Austria | Nancy Davolio | 12/11/19 | Federal Shipping | 96.43 | 2,216.25 |
| 10776 | ERNSH | Ernst Handel | Austria | Nancy Davolio | 12/15/19 | Federal Shipping | 351.53 | 6,984.50 |
| 10698 | ERNSH | Ernst Handel | Austria | Margaret Peacock | 10/09/19 | Speedy Express | 272.47 | 3,600.73 |
| 10747 | PICCO | Piccolo und mehr | Austria | Michael Suyama | 11/19/19 | Speedy Express | 117.33 | 1,912.85 |
| 10530 | PICCO | Piccolo und mehr | Austria | Janet Leverling | 05/08/19 | United Package | 339.22 | 4,150.00 |
| 10514 | ERNSH | Ernst Handel | Austria | Janet Leverling | 04/22/19 | United Package | 789.95 | 5,623.45 |
| 10489 | PICCO | Piccolo und mehr | Austria | Michael Suyama | 03/28/19 | United Package | 5.29 | 502.20 |
| 10442 | ERNSH | Ernst Handel | Austria | Janet Leverling | 02/11/19 | United Package | 47.94 | 1,792.00 |
| 10402 | ERNSH | Ernst Handel | Austria | Laura Callahan | 01/02/19 | United Package | 67.88 | 2,713.50 |
| 10403 | ERNSH | Ernst Handel | Austria | Margaret Peacock | 01/03/19 | Federal Shipping | 73.79 | 1,005.90 |
| 10427 | PICCO | Piccolo und mehr | Austria | Margaret Peacock | 01/27/19 | United Package | 31.29 | 651.00 |
| 10430 | ERNSH | Ernst Handel | Austria | Margaret Peacock | 01/30/19 | Speedy Express | 458.78 | 5,796.00 |
| 10571 | ERNSH | Ernst Handel | Austria | Janet Leverling | 06/17/19 | Federal Shipping | 25.06 | 947.75 |
| 10595 | ERNSH | Ernst Handel | Austria | Andrew Fuller | 07/10/19 | Speedy Express | 96.78 | 6,300.00 |
| 10597 | PICCO | Piccolo und mehr | Austria | Robert King | 07/11/19 | Federal Shipping | 35.12 | 800.10 |
| 10633 | ERNSH | Ernst Handel | Austria | Robert King | 08/15/19 | Federal Shipping | 477.90 | 6,483.05 |
| 10667 | ERNSH | Ernst Handel | Austria | Robert King | 09/12/19 | Speedy Express | 78.09 | 1,921.00 |
| 10686 | PICCO | Piccolo und mehr | Austria | Andrew Fuller | 09/30/19 | Speedy Express | 96.50 | 1,638.45 |
| Subtotal (Austria) : | | | | | | | 3,745.65 | 63,151.98 |
| 10649 | MAISD | Maison Dewey | Belgium | Steven Buchanan | 08/28/19 | Federal Shipping | 6.20 | 1,434.00 |
| 10458 | SUPRD | Suprêmes délices | Belgium | Robert King | 02/26/19 | Federal Shipping | 147.06 | 3,891.00 |
| 10463 | SUPRD | Suprêmes délices | Belgium | Steven Buchanan | 03/04/19 | Federal Shipping | 14.78 | 713.30 |
| 10475 | SUPRD | Suprêmes délices | Belgium | Anne Dodsworth | 03/14/19 | Speedy Express | 68.52 | 1,770.80 |
| 10529 | MAISD | Maison Dewey | Belgium | Steven Buchanan | 05/07/19 | United Package | 66.69 | 946.00 |
| 10760 | MAISD | Maison Dewey | Belgium | Margaret Peacock | 12/01/19 | Speedy Express | 155.64 | 3,304.00 |
| 10767 | SUPRD | Suprêmes délices | Belgium | Margaret Peacock | 12/05/19 | Federal Shipping | 1.59 | 28.00 |
| Subtotal (Belgium) : | | | | | | | 460.48 | 12,067.10 |
| 10777 | GOURL | Gourmet Lanchonetes | Brazil | Robert King | 12/15/19 | United Package | 3.01 | 260.00 |
| 10770 | HANAR | Hanari Carnes | Brazil | Laura Callahan | 12/09/19 | Federal Shipping | 5.32 | 315.00 |
| 10766 | QUEEN | Queen Cozinha | Brazil | Laura Callahan | 12/19/19 | Speedy Express | 110.87 | 1,913.85 |
| 10794 | QUEDE | Que Delicia | Brazil | Michael Suyama | 12/24/19 | Speedy Express | 21.49 | 393.45 |
| Total : | | | | | | | 32,468.77 | 658,388.75 |

By clicking on each column's heading ("1"), data display can be sort in ascending or descending order based on the clicked-heading.

By enabling the "Display Subtotal" checkbox ("2"), then clicking on a data column's heading will generate subtotal based on the clicked-heading, such as:

- Clicking on "Country", the display will group and generate subtotal for each country.
- Clicking on "CustomerID", the display will group and generate subtotal for each customer.
- Clicking on "SalesRep", the display will group and generate subtotal for each SalesRep.

To view only the subtotal without the details, select the "Hide Details" checkbox.

(Hiding details only work when "Display Subtotal" is enabled)

Filtering Data

Data can be filtered and limited within the specified From and To dates. In addition, the text-box above the column-heading can be used to enter a value to filter data (not all column can be used to filter data).



| OrderID | CustomerID | CompanyName | Country | SalesRep | OrderDate | Shipper | Freight | OrderTotal |
|---------|------------|--------------------|---------|-----------------|-----------|----------------|---------|------------|
| 10626 | BERGS | Berglunds snabbköp | Sweden | Nancy Davolio | 08/11/19 | United Package | 138.69 | 1503.60 |
| 10654 | BERGS | Berglunds snabbköp | Sweden | Steven Buchanan | 09/02/19 | Speedy Express | 55.26 | 668.70 |
| 10672 | BERGS | Berglunds snabbköp | Sweden | Anne Dodsworth | 09/17/19 | United Package | 95.75 | 4210.50 |
| Total : | | | | | | | 289.70 | 6382.80 |

Data filtering example:

- Above the "Country" column-heading, enter "SWEDEN" and press enter will filter and display data for Sweden.
- Change the From-date to "2019-8-1" and To-date to "2019-9-30" will further filter the data and only show orders between these two days

Note:

Click on the "Go" button to process data filtering.

When entering filter value in the column-heading, the enter key will trigger data filtering. If not, click on the "Go" button.

Excel Export

The "Excel export" will generate an Excel file based on the data currently showing on the web page.

The "Excel export" function only works for Webform, on a Windows machine.

For the CustomerOrderXml.html web application, "CSV export" to generate data in CSV format, which can be open in excel in less desirable display format comparing to the Excel export from a WebForm.

Note:

There are JavaScript libraries to generate Excel file.

These codes are lengthy and complex. To keep the code short and simple, CSV export is used.

CustomerOrderXml.html

The CustomerOrderXml.html web application is in HTML without using any framework.

An associated JavaScript file, CustomerOrderXml.js, is used to capture mouse-click event, request data from the SOAP web service, and display the XML data returned from the web service.

There are two additional JavaScript files in the \JS folder.

- The Utility.js file contains routine to retrieve node-name and node-value from the XML data.
- The HttpRequest.js file contains XMLHttpRequest() routine to request XML data from the SOAP web service.

In the CustomerOrderXml.js JavaScript file, the following block of code provides the mechanism to execute the intended JavaScript when the HTML content is loaded:

```
let PageMngr = null;

window.onload = window_onload;
function window_onload()
{
    try
    {
        PageMngr = new PageManager();
        PageMngr.DisplayInit();
        PageMngr.GetData();
    }
    catch (ex)
    {
        alert(ex.description);
        window.status = ex.description;
    }
}
```

The "PageMngr.DisplayInit()" calls the following function to initialize display:

```
this.DisplayInit = DisplayInit;
function DisplayInit()
{
    let _today = new Date();
    ToDate = _today.getFullYear() + "-" + (_today.getMonth() + 1) + "-" + _today.getDate();
    if (txtToDate) { txtToDate.value = ToDate; }

    let _fromDate = new Date();
    _fromDate.setMonth(_fromDate.getMonth() - 12);
    FromDate = _fromDate.getFullYear() + "-" + (_fromDate.getMonth() + 1) + "-" + _fromDate.getDate();
    if (txtFromDate) { txtFromDate.value = FromDate; }

    DisplaySubtotal = "0";
    chkDisplaySubTotals.checked = false;

    HideDetails = "0";
    chkHideDetails.checked = false;
}
```

The "PageMngr.GetData()" calls the following function to retrieve data from a web service:

```
let http = new HttpRequest();
this.GetData = GetData;
function GetData()
{
    TotalOrder = 0.0;
    TotalFreight = 0.0;

    let _arguments = "<_FromDate>" + FromDate + "</_FromDate>\n"
```

```

+ "<_ToDate>" + ToDate + "</_ToDate>\n"
+ "<_OrderBy>" + OrderBy + "</_OrderBy>\n"
+ "<_AscDesc>" + AscDesc + "</_AscDesc>\n"
+ "<_OrderID>" + OrderID + "</_OrderID>\n"
+ "<_CustomerID>" + CustomerID + "</_CustomerID>\n"
+ "<_CompanyName>" + CompanyName + "</_CompanyName>\n"
+ "<_Country>" + Country + "</_Country>\n"
+ "<_SalesRep>" + SalesRep + "</_SalesRep>\n"
+ "<_Shipper>" + Shipper + "</_Shipper>\n";
http.Execute("http://IotZones.com/", asmUrl, "Request", _arguments, onResponse);
}

function onResponse(RspNode, Errors)
{
    try
    {
        if (Errors) { alert(Errors); }
        else if (RspNode != null)
        {
            var OrderData = GetNodeByName(RspNode, "Rows");
            if (OrderData != null) { DisplayXmlData(OrderData); }
        }
    }
    catch (ex)
    {
        alert(ex.description, true);
    }
}

```

After retrieving data from the web service, the following function is called to display data to an HTML table:

```

let ReportBody = document.getElementById("ReportBody");
let ReportFooterBody = document.getElementById("ReportFooterBody");
function DisplayXmlData(_orderData)
{
    let _displayData = "";
    let _orderByValue = "";
    let _subTotalOrder = 0.0;
    let _subTotalFreight = 0.0;

    if (_orderData.childNodes.length > 0)
    {
        _orderByValue = GetNodeValueByName(_orderData.childNodes[0], "OrderByValue");
    }

    for (let ix = 0; ix < _orderData.childNodes.length; ix++)
    {
        var itm = _orderData.childNodes[ix];
        if (itm.nodeType == 1)
        {
            if (OrderBy != "" && chkDisplaySubTotals.checked)
            {
                if (_orderByValue == GetNodeValueByName(itm, "OrderByValue"))
                {
                    _subTotalOrder += parseFloat(GetNodeValueByName(itm, "OrderTotal").replace(/,/g, ''));
                    _subTotalFreight += parseFloat(GetNodeValueByName(itm, "Freight").replace(/,/g, ''));
                }
                else
                {
                    _displayData +=
                        "<tr class='rwSubtotal'><th colspan='7'>Subtotal: ( " + _orderByValue + " ): </th>"
                        + "<th>" + _subTotalFreight.toFixed(2) + "</th>"
                        + "<th>" + _subTotalOrder.toFixed(2) + "</th></tr>";

                    _subTotalOrder = parseFloat(GetNodeValueByName(itm, "OrderTotal").replace(/,/g, ''));
                    _subTotalFreight = parseFloat(GetNodeValueByName(itm, "Freight").replace(/,/g, ''));
                    _orderByValue = GetNodeValueByName(itm, "OrderByValue");
                }
            }
        }

        let _orderTotal = parseFloat(GetNodeValueByName(itm, "OrderTotal").replace(/,/g, ''));
        let _freight = parseFloat(GetNodeValueByName(itm, "Freight").replace(/,/g, ''));
    }
}

```

```

        if ((!chkHideDetails.checked) || (!chkDisplaySubTotals.checked))
        {
            _displayData += "<tr><td>" + GetNodeValueByName(itm, "OrderID") + "</td>"
                + "<td>" + GetNodeValueByName(itm, "CustomerID") + "</td>"
                + "<td>" + GetNodeValueByName(itm, "CompanyName") + "</td>"
                + "<td>" + GetNodeValueByName(itm, "Country") + "</td>"
                + "<td>" + GetNodeValueByName(itm, "SalesRep") + "</td>"
                + "<td>" + GetNodeValueByName(itm, "OrderDate") + "</td>"
                + "<td>" + GetNodeValueByName(itm, "Shipper") + "</td>"
                + "<td>" + _freight.toFixed(2) + "</td>"
                + "<td>" + _orderTotal.toFixed(2) + "</td></tr>";
        }
        TotalOrder += _orderTotal;
        TotalFreight += _freight;
    }
}

if (OrderBy != "" && chkDisplaySubTotals.checked)
{
    _displayData += "<tr class='rwSubtotal'><th colspan='7'>Subtotal: ( " + _orderByValue + " ): </th>"
        + "<th>" + _subTotalFreight.toFixed(2) + "</th>"
        + "<th>" + _subTotalOrder.toFixed(2) + "</th></tr>";
}

if (ReportBody)
{
    ReportBody.innerHTML = _displayData;
}

if (ReportFooterBody)
{
    let _footerData = "<tr class='rwFooter'><th>Total :</th>"
        + "<th>" + TotalFreight.toFixed(2) + "</th>"
        + "<th>" + TotalOrder.toFixed(2) + "</th></tr>";

    ReportFooterBody.innerHTML = _footerData;
}
}

```

Mouse Events

Mouse events are captured in different region of the web application, process by the following functions:

The following function process mouse events in the Form Header region:

```

let tblFormHeader = document.getElementById("tblFormHeader");
if (tblFormHeader) { tblFormHeader.onclick = tblFormHeader_onclick; }
function tblFormHeader_onclick(e)
{
    let evt = e || window.event;
    let elm = evt.srcElement || evt.target;
    if ((elm.id == "chkDisplaySubTotals") || (elm.id == "chkHideDetails"))
    {
        SubmitForm();
    }
}

```

The following function process mouse events in the data-column heading region:

```

let tblHeader = document.getElementById("tblReportHeader");
if (tblHeader) { tblHeader.onclick = tblHeader_onclick; }
function tblHeader_onclick(e)
{
    let evt = e || window.event;
    let elm = evt.srcElement || evt.target;
    if (elm.tagName == "A")
    {
        let OrderBy = elm.id.replace("OrderBy.", "");
        if (hdnOrderBy.value == OrderBy)
        {
            hdnAscDesc.value = (hdnAscDesc.value == "1") ? "" : "1";
        }
    }
}

```

```

    }
    else
    {
        hdnAscDesc.value = "";
        hdnOrderBy.value = OrderBy;
    }
    SubmitForm();
    return false;
}
return true;
}

```

The following function process mouse and keyboard events in the data-column heading 's filter region:

```

let Filters = document.getElementById("Filters");
if (Filters) { Filters.onkeypress = txt_keypress; }
function txt_keypress(e)
{
    let evt = e || window.event;
    let kc = evt.keyCode || evt.which;
    if (kc == 13)
    {
        SubmitForm();
        return false;
    }
    return true;
}

```

All mouse/keyboard event handler call the following function to refresh display:

```

function SubmitForm()
{
    let elms = document.getElementsByTagName("INPUT");
    for (let ix = 0; ix < elms.length; ix++)
    {
        let elm = elms[ix];
        if (elm.type == "text")
        {
            switch (elm.id)
            {
                case "txtFromDate":
                    FromDate = elm.value;
                    break;
                case "txtToDate":
                    ToDate = elm.value;
                    break;
                case "txtOrderID":
                    OrderID = elm.value;
                    break;
                case "txtCustomerID":
                    CustomerID = elm.value;
                    break;
                case "txtCompanyName":
                    CompanyName = elm.value;
                    break;
                case "txtCountry":
                    Country = elm.value;
                    break;
                case "txtSalesRep":
                    SalesRep = elm.value;
                    break;
                case "txtShipper":
                    Shipper = elm.value;
                    break;
            }
        }
        else if (elm.type == "hidden")
        {
            switch (elm.id)
            {
                case "hdnOrderBy":
                    OrderBy = elm.value;

```

```
        break;
    case "hdnAscDesc":
        AscDesc = elm.value;
        break;
    }
}
else if (elm.type == "checkbox")
{
    switch (elm.id)
    {
        case "chkDisplaySubTotals":
            DisplaySubtotal = elm.checked ? "1" : "";
            break;
        case "chkHideDetails":
            HideDetails = elm.checked ? "1" : "";
            break;
    }
}
}

GetData();
}
```