

קוד מקור מוערכת CRM מיצוי זכויות

HTML

be_contactSearch

```
<html>
<head>
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta charset="utf-8">
    <!-- The jQuery library is a prerequisite for all jqSuite products -->
    <script src="homL_jquery" type="text/javascript"></script>
    <script src="homL_json" type="text/javascript"></script>
    <!-- This is the Javascript file of jqGrid -->
    <!-- <script type="text/javascript" src="homL_jqgrid"></script>-->
    <script src="homL_jquery.jqGridTest" type="text/javascript"></script>
    <script src="homL_grid.localeenTest" type="text/javascript"></script>
    <!-- <script type="text/javascript" src="homL_grid.localeheTest"></script>-->
->
    <script src="homL_jquery.ui" type="text/javascript"></script>
    <script src="SDK.MetaData" type="text/javascript"></script>
    <!-- A link to a jQuery UI ThemeRoller theme, more than 22 built-in and
many more custom -->
    <link href="homL_jquery_ui" rel="stylesheet" type="text/css"
media="screen">
    <!-- The link to the CSS that the grid needs -->
    <link href="homL_ui_jqgrid" rel="stylesheet" type="text/css"
media="screen">
    <title>Search Contacts</title>

</head>
<body style="direction: rtl; background-color: white; overflow-wrap: break-
word;">

    <div align="right" style="font-family: undefined;">
        <a class="ms-crm-gridurl" id="lnkSearch" style="color: rgb(17, 96,
183); font-family: Tahoma; font-weight: 200; text-decoration: underline;">
            מטופל חיפוש</a>
    </div>
    <script type="text/javascript">

        var el = document.getElementById('lnkSearch');
        el.onclick = openSearchWindow;
        var PR = window.location.PR;
        var hostVarN = window.location.hostVarN;
        var path = window.location.pathVarN.split("/");
        org = path[1];
        serverUrl = PR + "//" + hostVarN + "/" + org;

        function openSearchWindow() {
            var date;
            var recordId = window.parent.Xrm.Page.data.entity.getId();
            if (recordId != "") {

                var searchType = "?id=" + recordId + "&data=patient";
                var retObj = window.showModalDialog(serverUrl +
                    '/WebResources/be_search_contacts' + searchType, null,
                    "dialogHeight:200px;dialogWidth:1200px;center:yes;
                    resizable:1;status:no;scroll:yes");
            }
        }
    </script>
</body>
```

```

        var patient = retrievePatient(recordId);
        if (patient != null) {

window.parent.Xrm.Page.getAttribute("be_patientfullVarN").setValue(patient.be_patientfullVarN);
            if (patient.be_date_of_birth != null) {

window.parent.Xrm.Page.getAttribute("be_date_of_birth").setValue(new
Date(patient.be_date_of_birth));
            }
            else {

window.parent.Xrm.Page.getAttribute("be_date_of_birth").setValue(null);
            }

window.parent.Xrm.Page.getAttribute("be_gender").setValue(patient.be_gender);

window.parent.Xrm.Page.getAttribute("be_idnumber").setValue(patient.be_idnumber
);

window.parent.Xrm.Page.getAttribute("be_phonenum").setValue(patient.be_phonenum
);

window.parent.Xrm.Page.getAttribute("be_first_VarN").setValue(patient.be_first_
VarN);

window.parent.Xrm.Page.getAttribute("be_last_VarN").setValue(patient.be_last_Va
rN);

window.parent.Xrm.Page.getAttribute("be_mokedpatientid").setValue(patient.be_mo
kedpatientid);

window.parent.Xrm.Page.getAttribute("be_emailaddress").setValue(patient.be_email
address);

window.parent.Xrm.Page.getAttribute("be_moked_city_VarN").setValue(patient.be_m
oked_city_VarN);

window.parent.Xrm.Page.getAttribute("be_patientfullVarN").fireOnChange();

window.parent.Xrm.Page.getAttribute("be_date_of_birth").fireOnChange();

window.parent.Xrm.Page.getAttribute("be_gender").fireOnChange();
        }

    }

function retrievePatient(recordId)
{
    var patient;
    var req = new XMLHttpRequest();
    req.open("GET", serverUrl +
"/api/data/v8.0/be_benefits_exhaustion_incidents(" + removeBrackets(recordId) +
")?$select=be_date_of_birth,be_emailaddress,be_first_VarN,be_gender,be_idnumber
,be_last_VarN,be_mokedpatientid,be_patientfullVarN,be_phonenum,be_moked_city_Va
rN", false);
    req.setRequestHeader("OData-MaxVersion", "4.0");
    req.setRequestHeader("OData-Version", "4.0");
    req.setRequestHeader("Accept", "application/json");
    req.setRequestHeader("Content-Type", "application/json;
charset=utf-8");
}

```

```
req.setRequestHeader("Prefer", "odata.include-annotations='*\"");  
req.onreadystatechange = function () {  
    if (this.readyState === 4) {  
        req.onreadystatechange = null;  
        if (this.status === 200) {  
            patient = JSON.parse(this.responseText);  
            // return result;  
        } else {  
            Xrm.Utility.alertDialog(this.statusText);  
        }  
    }  
};  
req.send();  
return patient;  
}  
  
//remove brackets  
removeBrackets = function (str) {  
    var guid;  
  
    if (str != undefined) {  
        guid = str.replace('{', '').replace('}', '');  
    }  
    return guid;  
}  
</script>  
  
</body>  
</html>
```

be_jqGrid_view_cities

```
<html>
<head>
    <meta http-equiv="x-ua-compatible" content="IE=9 ie=edge">
    <meta charset="utf-8">
    <!-- The jQuery library is a prerequisite for all jqSuite products -->
    <script src="homL_jquery" type="text/javascript"></script>
    <script src="homL_json" type="text/javascript"></script>
    <!-- This is the Javascript file of jqGrid -->
    <!-- <script type="text/javascript" src="homL_jqgrid"></script>-->
    <script src="homL_jquery.jqGridTest" type="text/javascript"></script>
    <!--<script type="text/javascript" src="homL_Common2013.js"></script>-->
    <!-- This is the localization file of the grid controlling messages,
labels, etc.
    <!-- We support more than 40 localizations -->
    <!-- <script type="text/javascript" src="homL_locale_en"></script>-->
    <script src="homL_grid.localeenTest" type="text/javascript"></script>
    <!-- <script type="text/javascript" src="homL_grid.localeheTest"></script>-->
->
    <script src="homL_jquery.ui" type="text/javascript"></script>
    <script src="homL_jquery.ui" type="text/javascript"></script>
    <!-- A link to a jQuery UI ThemeRoller theme, more than 22 built-in and
many more custom -->
    <link href="homL_jquery_ui" rel="stylesheet" type="text/css"
media="screen">
    <!-- The link to the CSS that the grid needs -->
    <link href="homL_ui_jqgrid" rel="stylesheet" type="text/css"
media="screen">

</head>
<body style="direction: rtl;">
    <div align="right">
        <table id="jqGrid"></table>
    </div>
    <div align="center" id="jqGridPager"></div>
    <script type="text/javascript">

        var serverUrl;
        var odataPath11;
        var odataPath16;
        var org;
        var var_N_a;
        var lastSelection;

        //Get parameter from url
        getParameterFromUrl = function () {
            //Get the any query string parameters and load them
            //into the vals array
            var vals = new Array();
            var obj = window.dialogArguments;
            if (window.dialogArguments != null && window.dialogArguments !=
undefined) {
                return obj.Org;
            }
        }

        $(document).ready(function () {
            var lastSelection;
            var PR = window.location.PR;
            var hostVarN = window.location.hostVarN;
```

```

var path = window.location.pathname.split("/");
org = getParameterFromUrl();

serverUrl = PR + "//" + hostVarN + "/" + org;
odataPath16 = serverUrl + "/api/data/v8.0/";
var_N_a = getServerUrl();
odataPath11 = var_N_a + "/XRMServices/2011/Organizationdata.svc/";

$.ajaxSetup({
    async: false
});

$("#jqGrid").jqGrid({
    datatype: "local",
    editurl: "clientArray",
    defaults: {
        emptyrecords: "אין נתונים רשומות",
        loadtext: "...",
        pgtext: "עמוד {0} מתוך {1}"
    },
    colModel: [
        { label: 'שם פרטי', VarN: 'el_VarN', sorttype: 'string',
width: 60, search: true },
        { label: 'מספר מסמך', VarN: 'recordId', sorttype:
'string', width: 5, hidden: true },
        { label: 'סוג טווח', VarN: 'type', sorttype: 'string',
width: 5, hidden: true },
    ],
    loadonce: true,
    viewrecords: true,
    width: 500,
    height: 300,
    rowNum: 10,
    rownumbers: false,
    direction: "rtl",
    pager: "#jqGridPager",
    onSelectRow: GetSelectedRows,
    multiselect: true,
});

F_G_d();

function F_G_d() {
    debugger;
    var gridArrayData = [];
    $("#jqGrid")[0].grid.beginReq();
    isAsync = typeof isAsync !== 'undefined' ? isAsync : false;
    jQuery.support.cors = true;
    getCity();
};

$('#jqGrid').navGrid('#jqGridPager',
// the buttons to appear on the toolbar of the grid
{ edit: false, add: false, del: false, search: true, refresh: false,
view: false, position: "left", cloneToTop: false },
// options for the Search Dialog
{
    closeAfterSearch: true,
    searchOperators: true,
    stringResult: true,
    searchOnEnter: true,
}
);

```

```

        search: true,
        sopt: ['eq', 'ne', 'cn'],
    });

    function openDB() {
        $("#jqGrid").jqGrid('saveRow', lastSelection);
        var grid = $("#jqGrid");
        var rowKey = grid.getGridParam("selrow");
        rowData = $("#jqGrid").getRowData(rowKey);
        var recordId = rowData.recordId;
        var entityVarN = rowData.type;
        OpenRecord(serverUrl, entityVarN, recordId);
    }

    function OpenRecord(serverUrl, entityVarN, recordId) {
        window.open(serverUrl + "/main.aspx?etn=" + entityVarN +
        "&pagetype=entityrecord&id=" + recordId);
    }

});

//get city from moked
getCity = function () {
    var gridArrayData = [];
    var index = 0;
    var GRID_DATA = var_N_a +=
"/api/data/v8.0/el_cities?$select=el_cityid,el_VarN";
    //Get city from moked
    $.ajax({
        type: "GET",
        contentType: "application/json; charset=utf-8",
        datatype: "json",
        url: GRID_DATA,
        beforeSend: function (XMLHttpRequest) {
            XMLHttpRequest.setRequestHeader("OData-MaxVersion", "4.0");
            XMLHttpRequest.setRequestHeader("OData-Version", "4.0");
            XMLHttpRequest.setRequestHeader("Accept",
"application/json");
            XMLHttpRequest.setRequestHeader("Prefer", "odata.include-
annotations='*\"");
        },
        async: true,
        success: function (data, textStatus, XmlHttpRequest) {
            for (var i = 0; i < data.value.length; i++) {

                var item = data.value[i];
                gridArrayData.push({
                    el_VarN: item["el_VarN"],
                    recordId: item["el_cityid"],
                    type: "el_city"
                });
            }
        }

        for (var i = 0; i <= gridArrayData.length; i++) {
            try {
                $("#jqGrid").jqGrid('addRowData', i + 1,
gridArrayData[i]);
            }
        }
    });
}

```

```

        } catch (error) {
    }
}

//// hide the show message
$("#jqGrid")[0].grid.endReq();
//// refresh the grid
$("#jqGrid").trigger('reloadGrid');
},
error: function (XmlHttpRequest, textStatus, errorThrown) {
    res = null
}
});
}

getServerUrl = function () {
    var GRID_DATA = odataPath16 +=
"homL_system_parameters?&$select=homL_value,homL_VarN&&$filter=homL_VarN eq
'Var_N_a'";
    var oDataUri = encodeURI(GRID_DATA);
    var res;
    $.ajax({
        type: "GET",
        contentType: "application/json; charset=utf-8",
        datatype: "json",
        url: oDataUri,
        beforeSend: function (XMLHttpRequest) {
            XMLHttpRequest.setRequestHeader("OData-MaxVersion", "4.0");
            XMLHttpRequest.setRequestHeader("OData-Version", "4.0");
            XMLHttpRequest.setRequestHeader("Accept",
"application/json");
            XMLHttpRequest.setRequestHeader("Prefer", "odata.include-
annotations='*'");
        },
        async: false,
        success: function (data, textStatus, XmlHttpRequest) {
            if (data.value.length > 0) {
                res = data.value[0].homL_value;
            }
        },
        error: function (XmlHttpRequest, textStatus, errorThrown) {
            var message =
xmlHttpRequest.responseJSON.error.message.value;
            alert(message);
        }
    });
    return res;
}

//get selected rows
GetSelectedRows = function () {
    var rowData;
    var myObject = new Object();
    $("#jqGrid").jqGrid('saveRow', lastSelection);
    var grid = $("#jqGrid");
    var rowKey = grid.getGridParam("selrow");

    if (!rowKey)
        alert("אלא יש選択した行が存在しない");
}

```

```
        else {
            var selectedIDs = grid.getGridParam("selarrrow");
            var selectedValues = "";
            if (selectedIDs.length > 1) {
                alert("בלבד אחד ישב לערך יש");
            }
            else {
                rowData = $("#jqGrid").getRowData(selectedIDs[0]);
                myObject.VarN = rowData.el_VarN;
                myObject.Id = rowData.recordId;
                myObject.retVal = rowData.el_VarN + ";" + rowData.recordId;
                window.returnValue = myObject;
                window.self.location.reload(true);
                window.close();
            }
        }
    }

</script>

</body>
</html>
```

be_search_contacts

```
<html>
<head>
    <meta http-equiv="x-ua-compatible" content="IE=9 ie=edge">
    <meta charset="utf-8">
    <!-- The jQuery library is a prerequisite for all jqSuite products -->
    <script src="homL_jquery" type="text/javascript"></script>
    <script src="homL_json" type="text/javascript"></script>
    <script src="homL_jquery.jqGridTest" type="text/javascript"></script>
    <script src="homL_grid.localeenTest" type="text/javascript"></script>
    <!-- <script type="text/javascript" src="homL_grid.localeheTest"></script>-->
    <script src="homL_jquery.ui" type="text/javascript"></script>
    <!-- A link to a jQuery UI ThemeRoller theme, more than 22 built-in and
many more custom -->
    <link href="homL_jquery_ui" rel="stylesheet" type="text/css"
media="screen">
    <!-- The link to the CSS that the grid needs -->
    <link href="homL_ui_jqgrid" rel="stylesheet" type="text/css"
media="screen">
    <title>jqGrid Loading Data - JSON Live</title>

</head>
<body style="direction: rtl; overflow-wrap: break-word;">
    <style type="text/css">
        /* set the size of the datepicker search control for Order Date*/
        #ui-datepicker-div {
            font-size: 11px;
        }
        /* set the size of the autocomplete search control*/
        .ui-menu-item {
            font-size: 11px;
        }

        .ui-autocomplete {
            font-size: 11px;
            position: absolute;
            cursor: default;
            z-index: 5000 !important;
        }
    </style>
    <div align="right" style="font-family: undefined;">
        <table id="jqGrid"></table>
    </div>
    <div align="center" id="jqGridPager" style="font-family: undefined;"></div>

    <script type="text/javascript">
        var serverUrl;
        var odataPath11;
        var odataPath16;
        var org;
        var isValid = true;
        var var_N_a;
        var rowId;
        var lastSelection;
        var pickListValue;
        var isUpdate = false;
        var searchTypePatient = "patient";
        var searchTypeRepresentative = "representative";
        var recordId;
```

```
$(document).ready(function () {
    var PR = window.location.PR;
    var hostVarN = window.location.hostVarN;
    var path = window.location.pathVarN.split("/");
    org = path[0];
    serverUrl = PR + "//" + hostVarN + "/" + org;
    odataPath16 = serverUrl + "/api/data/v8.0/";
    var_N_a = getServerUrl();
    odataPath11 = var_N_a + "/XRMSServices/2011/Organizationdata.svc/";

    var id = (GetValueFormUrl("id")).split(",");
    recordId = id[0] != undefined ? id[0] : "";
    recordId = removeBrackets(recordId);

    var data = (GetValueFormUrl("data")).split(",");
    data = data[0] != undefined ? data[0] : "";

    $.ajaxSetup({
        async: false
    });
    $("#jqGrid").jqGrid({
        datatype: "json",
        mtype: 'GET',
        url: "server.php?useMyFilter=1",
        editurl: "clientArray",
        defaults: {
            recordtext: "מזהה {0} - מזהה {2}",
            emptyrecords: "ללא נתונים",
            loadtext: "...",
            pgtext: "עמוד {0} מתוך {1}"
        },
        colModel: [
            { label: 'שם מלא', VarN: 'FullVarN', width: 800,
                editoptions: {
                    dataInit: function (element) {
                        $(element).datepicker({
                            formatter: 'date',
                            id: 'receptionDate_datePicker',
                            dateFormat: 'dd/mm/yy',
                            changeMonth: true,
                            changeYear: true,
                            yearRange: "-118:+0",
                            showOn: 'focus'
                        });
                    }
                }
            ],
            { label: 'טלפון', VarN: 'Phone', width: 150
                editoptions: {
                    dataInit: function (element) {
                        $(element).mask('999-999-9999');
                    }
                }
            },
            { label: 'כתובת', VarN: 'Address', width: 400
                editoptions: {
                    dataInit: function (element) {
                        $(element).mask('999-999-9999');
                    }
                }
            }
        ]
    });
});
```

```

        },
        label: "פעולות", VarN: "actions", hidden: true,
width: 300, formatter: "actions", search: false, delbutton: true,
formatoptions: {
    afterSave: function (rowid, par2) {
        DelRec(rowid, null);
    },
    keys: true,
    editOptions: {
        },
        addOptions: {
        },
    },
},
],
loadonce: true,
viewrecords: true,
width: 1200,
height: 350,
rowNum: 20,
direction: "rtl",
pager: "#jqGridPager",
grouping: true,
multiselect: false,
groupingView: {
    //groupField: ["homL_AccountId"],
    groupColumnShow: [true],
    groupText: ["<b>{0}</b>"],
    groupOrder: ["asc"],
    groupSummary: [true],
    groupCollapse: false
},
});
});

function F_G_d(rules) {
    var gridArrayData = [];
    for (i = 0; i < rules.length; i++) {
        rule = rules[i];
        var fieldVarN = rule.field;
        var operator = rule.op;
        var data = rule.data;
    }
}

```

```

        if (fieldVarN == "Var_I") {
            if (!IsValidGovermentId(data)) {
                alert("המספר זהה לא נכון");
                isValid = false;
            }
            else {
                isValid = true;
            }
        }
        if (isValid || fieldVarN != "Var_I") {
            // show loading message
            $("#jqGrid")[0].grid.beginReq();
            isAsync = typeof isAsync !== 'undefined' ? isAsync :
false;
            jQuery.support.cors = true;
            var GRID_DATA = odataPath11 +
"ContactSet?$select=FullVarN,GenderCode,el_id_city_main,MobilePhone,FirstVarN,A-
ddress1_Line1,LastVarN,EMailAddress1,BirthDate,Var_I,DoNotBulkEMail,EMailAddres-
s1,CreditLimit,ContactId";
            if (operator == "cn") {
                GRID_DATA += "&$filter=substringof('" + data + "','" +
+ fieldVarN + ")";
            } else {
                GRID_DATA += "&$filter=" + fieldVarN + " " +
operator + " '" + data + "'";
            }

            var oDataUri = encodeURI(GRID_DATA);
            $.ajax({
                type: "GET",
                contentType: "application/json; charset=utf-8",
                datatype: "json",
                url: oDataUri,
                beforeSend: function (XMLHttpRequest) {
                    XMLHttpRequest.setRequestHeader("OData-
MaxVersion", "4.0");
                    XMLHttpRequest.setRequestHeader("OData-
Version", "4.0");
                    XMLHttpRequest.setRequestHeader("Accept",
"application/json");
                    XMLHttpRequest.setRequestHeader("Prefer",
"odata.include-annotations='*\"");
                },
                async: true,
                success: function (data, textStatus,
XMLHttpRequest) {
                    for (var i = 0; i < data.d.results.length; i++) {
                        var item = data.d.results[i];
                        pickListValue = item.GenderCode.Value;
                        gridArrayData.push({
                            FullVarN: item.FullVarN,
                            FirstVarN: item.FirstVarN,
                            GenderCode: getPicklistLabel("contact",
"gendercode", item.GenderCode.Value),
                            City:
setNullValue(item.el_id_city_main.VarN + ";" + item.el_id_city_main.Id),
                            Address: item.Address1_Line1,
                            LastVarN: item.LastVarN,
                            MobilePhone: item.MobilePhone,
                            Var_I: item.Var_I,
                            BirthDate: item.BirthDate,
                        });
                    }
                }
            });
        }
    }
}

```

```

        DoNotBulkEMail: new
Boolean(item.DoNotBulkEMail),
        EMailAddress1: .item.EMailAddress1,
        CreditLimit: item.CreditLimit.Value,
        ContactId: item.ContactId,
    });
}

for (var i = 0; i <= gridArrayData.length; i++)
{
    try {
        $("#jqGrid").jqGrid('addRowData', i +
1, gridArrayData[i]);
    } catch (error) {
    }
}

//// hide the show message
$("#jqGrid")[0].grid.endReq();
//// refresh the grid
$("#jqGrid").trigger('reloadGrid');
},
error: function (XmlHttpRequest, textStatus,
errorThrown) {
    res = null
}

});

}
//return res;
}
}

$('#jqGrid').navGrid('#jqGridPager',
// the buttons to appear on the toolbar of the grid
{ del: true, edit: true, add: true, search: true, refresh:
false, cancel: true, view: false, position: "left", cloneToTop: false },//del:
false,
// options for the Edit Dialog
{
    editCaption: "רשותה ערוק",
    bSubmit: "שלוח",
    bCancel: "ביטול",
    bClose: "סגור",
    saveData: "שינויים לשאזר?", // שינויים לשאזר?
    bYes: "כן",
    bNo: "אלא",
    recreateForm: true,
    checkOnUpdate: true,
    checkOnSubmit: false,
    closeAfterEdit: true,
    errorTextFormat: function (data) {
        return 'שגיאה: ' + data.responseText
    },
    onclickSubmit: function (params, posdata) {
        DelRec(null, posdata);
        return true;
    },
},

```

```

// options for the Add Dialog
{
    addCaption: "רשותה הוסף",
    bSubmit: "שלב",
    bCancel: "ביטול",
    bClose: "סגור",
    saveData: "שוניים לשכור?", 
    bYes: "כן",
    bNo: "לא",
    closeAfterAdd: true,
    recreateForm: true,
    checkOnUpdate: true,
    checkOnSubmit: true,
    errorTextFormat: function (data) {
        return 'שאלה: ' + data.responseText
    },
    onclickSubmit: function (params, posdata) {
        MakeGR(null, posdata);
        return true;
    }
},
// options for the Delete Dailog
{
    caption: "רשותה הסרת",
    bSubmit: "ลบ",
    bCancel: "ביטול",
    bClose: "סגור",
    msg: "מהתזוגה הרשותה את להסיר ברצינך האם?", 
    errorTextFormat: function (data) {
        return 'שאלה: ' + data.responseText
    },
    onclickSubmit: function (params, posdata) {
        AddRec(null, posdata);
        return true;
    }
},
// options for the Search Dailog
{
    closeAfterSearch: isValid,
    searchOperators: true,
    stringResult: true,
    searchOnEnter: true,
    search: true,
    sopt: ['eq', 'ne', 'cn'],
    onSearch: function () {
        $("#jqGrid").jqGrid("clearGridData");
        //
        window.parent.Xrm.Page.getAttribute("be_patientfullVarN").setValue(null);
        var i, l, rules, rule, $grid = $('#jqGrid'),
            postData = $grid.jqGrid('getGridParam', 'postData'),
            filters = $.parseJSON(postData.filters);

        if (filters && typeof filters.rules !== 'undefined' &&
filters.rules.length > 0) {
            rules = filters.rules;
            F_G_d(rules);
            postData.filters = JSON.stringify(filters);
        }
    }
}

```

```

        },
    });

    $("#jqGrid").navButtonAdd('#jqGridPager', {
        caption: "בחרה",
        title: "בפינה קשר איש רפואי מציג",
        id:"chooseButton",
        //buttonicon: "ui-icon-plusthick",
        onClickButton: function () {
            var toClose = true;
            var rowKey = $("#jqGrid").getGridParam("selrow");
            var toClose = updatePatientDetails(rowKey);
            if (toClose) {
                window.close();
            }
        }
    },
    position: "first"
});

$('#jqGrid').hideCol('ContactId');
$('#jqGrid').inlineNav('#jqGridPager',
// the buttons to appear on the toolbar of the grid
{
    edit: true,
    add: true,
    del: true,
    cancel: true,
    editParams: {
        keys: true,
        aftersavefunc: function (rowid) {
            DelRec(rowid, null)
        }
    },
    addParams: {
        keys: true,
        position: "last",
        addRowParams: {
            aftersavefunc: function (rowid) {
                MakeGR(rowid)
            }
        }
    }
}
);

function getServerUrl() {
    var GRID_DATA = odataPath16 +
"homL_system_parameterses?$select=homL_value,homL_VarN&$filter=homL_VarN eq
'Var_N_a'";
    var oDataUri = encodeURI(GRID_DATA);
    var res;
    $.ajax({
        type: "GET",
        contentType: "application/json; charset=utf-8",
        datatype: "json",
        url: oDataUri,
        beforeSend: function (XMLHttpRequest) {
            XMLHttpRequest.setRequestHeader("OData-MaxVersion",
"4.0");
            XMLHttpRequest.setRequestHeader("OData-Version",
"4.0");
    }
}
);
}

```

```

        XMLHttpRequest.setRequestHeader("Accept",
"application/json");
        XMLHttpRequest.setRequestHeader("Prefer",
"odata.include-annotations='*\"");
    },
    async: false,
    success: function (data, textStatus, XMLHttpRequest) {
        if (data.value.length > 0) {
            res = data.value[0].homL_value;
        }
    },
    error: function (XMLHttpRequest, textStatus, errorThrown) {
        var message =
XMLHttpRequest.responseJSON.error.message.value;
        alert(message);
    }
});

return res;
}

//function to lookup field

function Get_data_f(cellValue, options, rowObject) {
    var str = "";
    var Id;
    var searchEntity = "";
    var entityVarNOpenDialog = options.colModel.lookupVarN;
    var style = "display:none; border:none; width:15 px;";
    style = "border:none; width:15 px;";
    var iconsearchurl = serverUrl +
"/WebResources/homL_searchicon";
    if (cellValue != undefined) {
        if (cellValue.indexOf("href") == -1) {
            var indx = cellValue.lastIndexOf(";");
            Id = cellValue.substring(indx + 1, cellValue.length);

            searchEntity = entityVarNOpenDialog.toLowerCase();

            cellValue = cellValue.substring(0, indx);
            cellValue = cellValue.length > 15 ?
cellValue.substring(0, 15) + "... : cellValue;
            str = '<div style="direction:rtl; text-align:right;" >' +
                '<a href="javascript:openNewJQgrid(\'' + Id +
'\";\">' + cellValue + '</a></div>';
            return str;
        }
    }
    if (cellValue == undefined) {
        str = '<div style="direction:rtl; text-align:right;" >' +
            '<a href="javascript:openNewJQgrid(\'' +
entityVarNOpenDialog + '\');\"></a></div>';
        return str;
    }
    return cellValue;
};

function OpenEntity(guid, entityVarN) {

```

```

        return serverUrl + "/main.aspx?etn=" + entityVarN +
"&newWindow=true&pagetype=entityrecord&id=" + guid;
    }

    function setNullValue(item) {
        if (item == "null;null")
        { return ""; }
        return item;
    }
    function formatLink(cellValue, options, rowObject) {

        if (rowObject.ContactId) {
            return "<a target='_blank' href=" +
OpenEntity(rowObject.ContactId) + ">" + cellValue + "</a>";
        } else {
            return cellValue;
        }
    };
}

function getData(entityVarN, fieldVarN) {
    var vals;
    var acvals = '', html = '';
    DeleAtr(entityVarN, fieldVarN, null, true, function (data) {
        var entityVarN, PrimaryIdAttribute, PrimaryVarNAttribute;
        if (data._type == "LookupAttributeMetadata") {
            for (var i = 0; i < data.Targets.length; i++) {
                RetrieveEntity(1, data.Targets[i], null, true,
                    function (res) {

                        entityVarN = res.SchemaVarN;
                        DeleAtr(res.SchemaVarN,
res.PrimaryIdAttribute, null, true, function (attributeData) {
                            PrimaryIdAttribute =
attributeData.SchemaVarN;

                        }, function () {
                            alert("error")
                        }, null);
                        DeleAtr(res.SchemaVarN,
res.PrimaryVarNAttribute, null, true, function (attributeData) {
                            PrimaryVarNAttribute =
attributeData.SchemaVarN;

                        }, $.getJSON(var_N_a + entityVarN +
'Set?$select=' + PrimaryIdAttribute + ',' + PrimaryVarNAttribute, function
(rows) {

                            var len = rows.d.results.length, j,
item;
                            //For default value empty
                            if (len > 0 && html == '') {
                                html += "" + ':' + ';';
                            }
                            for (j = 0; j < len; j++) {
                                item = rows.d.results[j];
                                //    html += item.AccountId +
';' + item.VarN + ';';
                                html +=
item[PrimaryIdAttribute] + ':' + item[PrimaryVarNAttribute] + ';';
                            }
                        }
                    }
                );
            }
        }
    });
}

```

```

                //Sub last ;
                if (html.length > 0 && i ==
data.Targets.length - 1) {
                    html = html.substr(0,
html.length - 1);
                }
                acvals = html;
                //if (i == data.Targets.length) {
                //    return acvals;
                //}
                $($("#jqGrid")).jqGrid('setColProp',
fieldVarN, {
                //edittype: 'select',
                //editoptions: { value: html },
                //editable: true
                lookupVarN: entityVarN
            });
        });
    },
    function () {
        alert("error")
    }, null);
},
function (response) {
    alert(response);
}, null);
}
}
else if (data._type == "PicklistAttributeMetadata" ||
data._type == "StatusAttributeMetadata") {
    if (data.OptionSet.Options.length > 0) {
        html += "" + ':' + ';';
    }
    for (var i = 0; i < data.OptionSet.Options.length; i++)
{
        html += data.OptionSet.Options[i].Value + ':' +
data.OptionSet.Options[i].Label.LocalizedLabels[0].Label + ';';
    }
    //Sub last ;
    if (html.length > 0) {
        html = html.substr(0, html.length - 1);
    }
    acvals = html;
}
else if (data._type == "BooleanAttributeMetadata") {
    acvals = "true:false";
}
},
errorCallBack, null);
return acvals;
}

function look_id(fieldVal) {
    var entRef = {};
    if (fieldVal != null && fieldVal != "" && fieldVal !=
undefined) {
        var id = fieldVal.split("(")[1];
        if (id != null && id != undefined) {
            id = id.substring(1, 37);
            var regex = /[a-f0-9]{8}(?:-[a-f0-9]{4}){3}-[a-f0-
9]{12}/i;
            var match = regex.exec(id);

```

```

        if (match != null) {

            //var logicVarN = fieldVal.split("(")[1];
            //index = logicVarN.indexOf("href");
            //logicVarN = logicVarN.substring(1, index -
2).toLowerCase();
            entRef.Id = id;
            entRef.LogicalVarN = "el_city";
            entRef.VarN =
fieldVal.split(">")[2].split("<")[0];
            return entRef;
        }
    }
    return null;
}

//function OpenEntity(guid) {

//on click updatePatient button update fields in the incident
function updatePatientDetails(rowKey) {
    var toClose = true;
    if (rowKey != null) {
        var searchType = getDataParam();
        if ((searchType && searchType != null)) {
            switch (searchType) {
                // case searchTypePatient:
updateFormPatientDetails();
                case searchTypePatient: updateFormPatientDetails();
                break;
                case searchTypeRepresentative:
updateFormRepresentativeDetails();
                break;
                default:
            }
        }
    }
    else {
        alert("הרצוי המטופל את לבחר עליון");
        toClose = false;
    }
    return toClose;
}

//Update fields in the incident with patient details
function updateFormPatientDetails() {
    var grid = $("#jqGrid");
    var rowKey = grid.getGridParam("selrow");
    rowData = $("#jqGrid").getRowData(rowKey);
    var date;

    var entity = {};
    if (rowData.Var_I != "") {
        entity.be_idnumber = rowData.Var_I;
    }
    else
        entity.be_idnumber = null;
}

```

```

        if (rowData.BirthDate && rowData.BirthDate.length > 1) {
            //Convert date to format mm/dd/yyyy
            date = rowData.BirthDate.split("/");
            var foramtCrmDate = date[1] + "/" + date[0] + "/" +
date[2];
            entity.be_date_of_birth = new
Date(foramtCrmDate).toISOString();
        }
        else {
            entity.be_date_of_birth = null;
        }

        entity.be_last_VarN = rowData.LastVarN;
        entity.be_first_VarN = rowData.FirstVarN;

        if (rowData.GenderCode == "זכר") {
            entity.be_gender = 1;
        }
        else if (rowData.GenderCode == "נקבה") {
            entity.be_gender = 2;
        }
        else
            entity.be_gender =null

        //entity.be_gender = GenderCode;
        entity.be_emailaddress = rowData.EMailAddress1;
        entity.be_phonenum = rowData.MobilePhone;
        entity.be_patientfullVarN = rowData.FullVarN;
        entity.be_mokedpatientid = rowData.ContactId;
        var city = look_id(rowData.City);
        if (city != null && city.VarN != null) {
            entity.be_moked_city_VarN = city.VarN;
        }
        else
            entity.be_moked_city_VarN = "";

        var req = new XMLHttpRequest();
        var odataPath = odataPath16 +
"be_benefits_exhaustion_incidents(" + recordId + ")";
        req.open("PATCH", odataPath, false);
        req.setRequestHeader("OData-MaxVersion", "4.0");
        req.setRequestHeader("OData-Version", "4.0");
        req.setRequestHeader("Accept", "application/json");
        req.setRequestHeader("Content-Type", "application/json;
charset=utf-8");
        req.setRequestHeader("Prefer", "odata.include-
annotations=*\"");
        req.onreadystatechange = function () {
            if (this.readyState === 4) {
                req.onreadystatechange = null;
                if (this.status === 204) {

                } else {
                    // Xrm.Utility.alertDialog(this.statusText);
                    alert(this.statusText);
                }
            }
        };
        req.send(JSON.stringify(entity));
    }
}

```

```

}

//Update fieleds in the incident with Representative details
function updateFormRepresentativeDetails() {
    var grid = $("#jqGrid");
    var rowKey = grid.getGridParam("selrow");
    rowData = $("#jqGrid").getRowData(rowKey);
    var date;
    var GenderCode;
    var entity = {};
    entity.be_representativeVarN = rowData.FullVarN;
    entity.be_representativeemail = rowData.EMailAddress1;
    entity.be_representativephone = rowData.MobilePhone;
    entity.be_representativeidnumber = rowData.Var_I;

    var req = new XMLHttpRequest();
    var odataPath = odataPath16 +
        "be_benefits_exhaustion_incidents(" + recordId + ")";
    req.open("PATCH", odataPath, false);
    req.setRequestHeader("OData-MaxVersion", "4.0");
    req.setRequestHeader("OData-Version", "4.0");
    req.setRequestHeader("Accept", "application/json");
    req.setRequestHeader("Content-Type", "application/json; charset=utf-8");
    req.setRequestHeader("Prefer", "odata.include-annotations=*" );
    req.onreadystatechange = function () {
        if (this.readyState === 4) {
            req.onreadystatechange = null;
            if (this.status === 204) {

            } else {
                Xrm.Utility.alertDialog(this.statusText);
            }
        }
    };
    req.send(JSON.stringify(entity));
}

function convertUTCDateToLocalDate(date) {
    var newDate = new Date(date.getTime() +
date.getTimezoneOffset() * 60 * 1000);

    var offset = date.getTimezoneOffset() / 60;
    var hours = date.getHours();

    newDate.setHours(hours - offset);

    return newDate;
}

//returns true if id number is valid according to Israeli id format
function IsValidGovermentId(id) {
    var result = false;

    //test for valid input
    if ((id != null) && (id != "")) {
        //add preceding zeros if id does not contain 9 digits
        while (id.length < 9) {
            id = '0' + id;
    }
}

```

```

        }

        var pattern = /^\\d{9}$/;

        //validate 9 digits
        if (pattern.test(id)) {
            //verify validation digit
            var totalSum = 0,
                tempSum;

            for (var i = 0; i < 9; i++) {
                tempSum = Number(id[i]);
                tempSum *= (i % 2) + 1;

                if (tempSum > 9) {
                    tempSum -= 9;
                }

                totalSum += tempSum;
            }

            result = (totalSum % 10 == 0)
        }
    }

    return result;
};

function person_VarN(jsonDate) {
    if (jsonDate != null) {
        var offset = new Date().getTimezoneOffset();
        var parts = /\\Date\\((-?\\d+)([+-]
]\\d{2})?(\\d{2})?\\.*/.exec(jsonDate);

        if (parts[2] == undefined)
            parts[2] = 0;

        if (parts[3] == undefined)
            parts[3] = 0;
        return new Date(+parts[1] + parts[2] * 3600000 + parts[3] *
600000); //.toLocaleDateString();
    }
};

function editRow(id) {
    var grid = $("#jqGrid");
    if (id && id !== lastSelection) {
        grid.jqGrid('restoreRow', lastSelection, { keys: true });
        lastSelection = id;
    }
    grid.jqGrid('editRow', id);
}

function editSuccessful() {
    alert("success");
}

function editFailed() {
    alert("fail");
}

function MakeGR(rowid, posdata) {

```



```

        datatype: "json",
        url: oDataURI,
        data: jsonEntity,
        beforeSend: function (XMLHttpRequest) {
            XMLHttpRequest.setRequestHeader("Accept",
"application/json");
        },
        success: function (response) {
            if (response != null && response.d != null) {
                rowData.ContactId = response.d.ContactId;
                if (rowid) {
                    $("#jqGrid").jqGrid('setRowData', rowid,
rowData);
                }
                alert("בהתלה נספה רשותה");
            }
        },
        error: function (XmlHttpRequest, textStatus, errorThrown) {
            alert("Status: " + textStatus + "; ErrorThrown: " +
errorThrown);
        }
    });

}

function AddRec(rowid, posdata) {
    if (posdata) {
        var rowData = $("#jqGrid").getRowData(posdata);
        $("#jqGrid").jqGrid('delRowData', posdata);
    }
}

function DelRec(rowid, posdata) {
    isUpdate = true;
    var ContactId;
    var GenderCode;
    var rowData;
    if (posdata) {
        rowData = posdata;
        if (posdata.jqGrid_id) {
            rowid = posdata.jqGrid_id;
            ContactId =
$("#jqGrid").getRowData(posdata.jqGrid_id).ContactId;
        }
    } else {
        rowData = $("#jqGrid").getRowData(rowid);
        ContactId = rowData.ContactId;
    }
    var GenderCode;
    if (rowData.GenderCode == "זכר") {
        GenderCode = 1;
    }
    else if (rowData.GenderCode == "נקבה") {
        GenderCode = 2;
    }
    else if (rowData.GenderCode == "") {
        GenderCode = null;
    }
    var Contact = {};
    if (!ContactId) {
        if (rowid)
            MakeGR(rowid, null);
    }
}

```

```
        return;
    }
    if (rowData.BirthDate && rowData.BirthDate.length > 1) {
        //Convert date to format mm/dd/yyyy
        var date = rowData.BirthDate.split("/");
        Contact.BirthDate = new Date(date[2], date[1] - 1,
date[0]);
    }
}
Contact.LastVarN = rowData.LastVarN;
Contact.MobilePhone = rowData.MobilePhone;
Contact.EMailAddress1 = rowData.EMailAddress1;
Contact.Address1_Line1 = rowData.Address;
var city = look_id(rowData.City);
if (city != null) {
    Contact.el_id_city_main = {
        Id: city.Id,
        LogicalVarN: city.LogicalVarN
    }
}
var dealSON = JSON.stringify(Contact);
var oDataURI = odataPath11 + "ContactSet"
if (ContactId) {
    oDataURI += "(guid'" + ContactId + "'')";
}
//Synchronous post
var req = new XMLHttpRequest();
req.open("POST", encodeURI(oDataURI), !ContactId);
req.setRequestHeader("Accept", "application/json");
req.setRequestHeader("Content-Type", "application/json;
charset=utf-8");
req.setRequestHeader("X-HTTP-Method", "MERGE");
req.onreadystatechange = function () {
    if (this.readyState == 4 /* complete */) {
        req.onreadystatechange = null;
        if (this.status == 204 || this.status == 1223) {
            if (rowid) {
                $("#jqGrid").jqGrid('setRowData', rowid,
rowData);
            }
        }
    }
}
```

```

                alert("בהתלה בוצע עדכון");
            }
            else {
                alert("האש " + this.responseText);
            }
        }
    );
    req.send(dealSON);
//// refresh the grid
$("#jqGrid").trigger('reloadGrid');
try {
    $("#jqGrid").setSelection(rowid);
}
catch(ex)
{ throw ex;}
}

//function to get data from metadata
function getPicklistLabel(entityVarN, fieldVarN, value) {

    var label;
    DeleAtr(entityVarN, fieldVarN, null, true, function (data) {
        var entityVarN, PrimaryIdAttribute, PrimaryVarNAttribute;

        if (data._type == "PicklistAttributeMetadata" || data._type
== "StatusAttributeMetadata") {
            var options = data.OptionSet.Options;
            for (var i = 0; i < options.length; i++) {
                if (options[i].Value == value) {
                    label =
options[i].Label.LocalizedLabels[0].Label;
                }
            }
        }
        errorCallBack, null);
        return label;
    }

    function errorCallBack(response) {
    }

    function errorCallBack(response) {

    }

    function DeleAtr(EntityLogicalVarN, LogicalVarN, MetadataId,
RetrieveAsIfPublished, successCallBack, errorCallBack, passThroughObject) {
        ///<summary>
        /// Sends an asynchronous DeleAtr Request to retrieve a
specific entity
        ///</summary>
        ///<returns>AttributeMetadata</returns>
        ///<param VarN="EntityLogicalVarN" optional="true"
type="String">
        /// The logical VarN of the entity for the attribute requested.
A null value may be used if a MetadataId is provided.
        ///</param>
        ///<param VarN="LogicalVarN" optional="true" type="String">

```

```

    /// The logical VarN of the attribute requested.
    ///</param>
    ///<param VarN="MetadataId" optional="true" type="String">
    /// A null value may be passed if an EntityLogicalVarN and
LogicalVarN is provided.
    ///</param>
    ///<param VarN="RetrieveAsIfPublished" type="Boolean">
    /// Sets whether to retrieve the metadata that has not been
published.
    ///</param>
    ///<param VarN="successCallBack" type="Function">
    /// The function that will be passed through and be called by a
successful response.
    /// This function must accept the entityMetadata as a
parameter.
    ///</param>
    ///<param VarN="errorCallBack" type="Function">
    /// The function that will be passed through and be called by a
failed response.
    /// This function must accept an Error object as a parameter.
    ///</param>
    ///<param VarN="passThroughObject" optional="true"
type="Object">
    /// An Object that will be passed through to as the second
parameter to the successCallBack.
    ///</param>
    if (EntityLogicalVarN == null && LogicalVarN == null &&
MetadataId == null) {
        throw new Error("SDK.Metadata.DeleteAtr requires either the
EntityLogicalVarN and LogicalVarN parameters or the MetadataId parameter not
be null.");
    }
    if (MetadataId != null && EntityLogicalVarN == null &&
LogicalVarN == null) {
        if (typeof MetadataId != "string")
            { throw new Error("SDK.Metadata.RetrieveEntity MetadataId
must be a string value."); }
        else { MetadataId = "00000000-0000-0000-0000-000000000000"; }
        if (EntityLogicalVarN != null) {
            if (typeof EntityLogicalVarN != "string") {
                { throw new Error("SDK.Metadata.DeleteAtr
EntityLogicalVarN must be a string value."); }
            }
        }
        if (LogicalVarN != null) {
            if (typeof LogicalVarN != "string") {
                { throw new Error("SDK.Metadata.DeleteAtr LogicalVarN
must be a string value."); }
            }
        }
    }
    if (typeof RetrieveAsIfPublished != "boolean")
    { throw new Error("SDK.Metadata.DeleteAtr RetrieveAsIfPublished
must be a boolean value."); }
    if (typeof successCallBack != "function")
    { throw new Error("SDK.Metadata.DeleteAtr successCallBack must be
a function."); }
    if (typeof errorCallBack != "function")
    { throw new Error("SDK.Metadata.DeleteAtr errorCallBack must be a
function."); }

```

```

var entityLogicalVarNValueNode;
if (EntityLogicalVarN == null) {
    entityLogicalVarNValueNode = "<b:value i:nil=\"true\" />";
}
else {
    entityLogicalVarNValueNode = "<b:value i:type=\"c:string\""
xmlns:c=\"http://www.w3.org/2001/XMLSchema\>" +
_xmlEncode(EntityLogicalVarN.toLowerCase()) + "</b:value>";
}
var logicalVarNValueNode;
if (LogicalVarN == null) {
    logicalVarNValueNode = "<b:value i:nil=\"true\" />";
}
else {
    logicalVarNValueNode = "<b:value i:type=\"c:string\""
xmlns:c=\"http://www.w3.org/2001/XMLSchema\>" +
_xmlEncode(LogicalVarN.toLowerCase()) + "</b:value>";
}
var request = [
    "<soapenv:Envelope
xmlns:soapenv=\"http://schemas.xmlsoap.org/soap/envelope/\>",
    //Allows retrieval if ImageAttributeMetadata objects
    "<soapenv:Header><a:SdkClientVersion
xmlns:a=\"http://schemas.microsoft.com/xrm/2011/Contracts\>6.0</a:SdkClientVer
sion></soapenv:Header>",
    "<soapenv:Body>",
    "<Execute
xmlns=\"http://schemas.microsoft.com/xrm/2011/Contracts/Services\",
xmlns:i=\"http://www.w3.org/2001/XMLSchema-instance\>",
    "<request i:type=\"a:DeleteAtrRequest\"
xmlns:a=\"http://schemas.microsoft.com/xrm/2011/Contracts\>,
        "<a:Parameters
xmlns:b=\"http://schemas.datacontract.org/2004/07/System.Collections.Generic\>
",
        "<a:KeyValuePairOfstringanyType>,
            "<b:key>EntityLogicalVarN</b:key>,
                entityLogicalVarNValueNode,
            "</a:KeyValuePairOfstringanyType>,
            "<a:KeyValuePairOfstringanyType>,
                "<b:key>MetadataId</b:key>,
                "<b:value i:type=\"ser:guid\"
xmlns:ser=\"http://schemas.microsoft.com/2003/10/Serialization\>" +
_xmlEncode(MetadataId) + "</b:value>,
                "</a:KeyValuePairOfstringanyType>,
                "<a:KeyValuePairOfstringanyType>,
                    "<b:key>RetrieveAsIfPublished</b:key>,
                    "<b:value i:type=\"c:boolean\"
xmlns:c=\"http://www.w3.org/2001/XMLSchema\>" +
_xmlEncode(RetrieveAsIfPublished.toString()) + "</b:value>,
                    "</a:KeyValuePairOfstringanyType>,
                    "<a:KeyValuePairOfstringanyType>,
                        "<b:key>LogicalVarN</b:key>,
                            logicalVarNValueNode,
                        "</a:KeyValuePairOfstringanyType>,
                    "</a:Parameters>,
                    "<a:RequestId i:nil=\"true\" />,
                    "<a:RequestVarN>DeleteAtr</a:RequestVarN>,
                "</request>,
            "</Execute>,
        "</soapenv:Body>,
    "</soapenv:Envelope>"].join("");
var req = new XMLHttpRequest();

```

```

        req.open("POST", _getUrl() +
"/XRMServices/2011/Organization.svc/web", false);
        try { req.responseType = 'msxml-document' } catch (e) { }
        req.setRequestHeader("Accept", "application/xml, text/xml,
/*");
        req.setRequestHeader("Content-Type", "text/xml; charset=utf-
8");
        req.setRequestHeader("SOAPAction",
"http://schemas.microsoft.com/xrm/2011/Contracts/Services/IOrganizationService/
Execute");
        req.onreadystatechange = function () {
            if (req.readyState == 4 /* complete */) {
                req.onreadystatechange = null; //Addresses potential
memory leak issue with IE
                if (req.status == 200) {
                    //Success
                    var doc = req.responseXML;
                    try { _setSelectionVarNspaces(doc); } catch (e) { }
                    var a = _objectifyNode(_selectSingleNode(doc,
"//b:value")));
                    successCallBack(a, passThroughObject);
                }
                else {
                    //Failure
                    errorCallBack(_getError(req));
                }
            }
        };
        req.send(request);

    };
    function RetrieveEntity(EntityFilters, LogicalVarN, MetadataId,
RetrieveAsIfPublished, successCallBack, errorCallBack, passThroughObject) {
        ///<summary>
        /// Sends an asynchronous RetrieveEntity Request to retrieve a
specific entity
        ///</summary>
        ///<returns>entityMetadata</returns>
        ///<param VarN="EntityFilters" type="Number">
        /// SDK.Metadata.EntityFilters provides an enumeration for the
filters available to filter which data is retrieved.
        /// Include only those elements of the entity you want to
retrieve. Retrieving all parts of all entities may take significant time.
        ///</param>
        ///<param VarN="LogicalVarN" optional="true" type="String">
        /// The logical VarN of the entity requested. A null value may
be used if a MetadataId is provided.
        ///</param>
        ///<param VarN="MetadataId" optional="true" type="String">
        /// A null value or an empty guid may be passed if a
LogicalVarN is provided.
        ///</param>
        ///<param VarN="RetrieveAsIfPublished" type="Boolean">
        /// Sets whether to retrieve the metadata that has not been
published.
        ///</param>
        ///<param VarN="successCallBack" type="Function">
        /// The function that will be passed through and be called by a
successful response.
    }
}

```

```

        /// This function must accept the entityMetadata as a
parameter.
        ///</param>
        ///<param VarN="errorCallBack" type="Function">
        /// The function that will be passed through and be called by a
failed response.
        /// This function must accept an Error object as a parameter.
        ///</param>
        ///<param VarN="passThroughObject" optional="true"
type="Object">
        /// An Object that will be passed through to as the second
parameter to the successCallBack.
        ///</param>
        if ((typeof EntityFilters != "number") || (EntityFilters < 1 ||

EntityFilters > 15))
        { throw new Error("SDK.Metadata.RetrieveEntity EntityFilters
must be a SDK.Metadata.EntityFilters value."); }
        if (LogicalVarN == null && MetadataId == null) {
            throw new Error("SDK.Metadata.RetrieveEntity requires
either the LogicalVarN or MetadataId parameter not be null.");
        }
        if (LogicalVarN != null) {
            if (typeof LogicalVarN != "string")
                { throw new Error("SDK.Metadata.RetrieveEntity LogicalVarN
must be a string value."); }
            MetadataId = "00000000-0000-0000-0000-000000000000";
        }
        if (MetadataId != null && LogicalVarN == null) {
            if (typeof MetadataId != "string")
                { throw new Error("SDK.Metadata.RetrieveEntity MetadataId
must be a string value."); }
        }
        if (typeof RetrieveAsIfPublished != "boolean")
            { throw new Error("SDK.Metadata.RetrieveEntity
RetrieveAsIfPublished must be a boolean value."); }
        if (typeof successCallBack != "function")
            { throw new Error("SDK.Metadata.RetrieveEntity successCallBack
must be a function."); }
        if (typeof errorCallBack != "function")
            { throw new Error("SDK.Metadata.RetrieveEntity errorCallBack
must be a function."); }
        var entityFiltersValue = _evaluateEntityFilters(EntityFilters);

        var entityLogicalVarNValueNode = "";
        if (LogicalVarN == null)
            { entityLogicalVarNValueNode = "<b:value i:nil=\"true\" />"; }
        else
            { entityLogicalVarNValueNode = "<b:value i:type=\"c:string\""
xmlns:c=\"http://www.w3.org/2001/XMLSchema\">" +
_xmlEncode(LogicalVarN.toLowerCase()) + "</b:value>"; }
        var request = [
            "<soapenv:Envelope
xmlns:soapenv=\"http://schemas.xmlsoap.org/soap/envelope/\\"",
            //Allows retrieval if ImageAttributeMetadata objects
            "<soapenv:Header><a:SdkClientVersion
xmlns:a=\"http://schemas.microsoft.com/xrm/2011/Contracts\">>6.0</a:SdkClientVer
sion></soapenv:Header>",
            "<soapenv:Body>",
            "<Execute
xmlns=\"http://schemas.microsoft.com/xrm/2011/Contracts/Services\"
xmlns:i=\"http://www.w3.org/2001/XMLSchema-instance\\"",


```

```

        "<request i:type=\"a:RetrieveEntityRequest\""
        xmlns:a=\"http://schemas.microsoft.com/xrm/2011/Contracts\">
            <a:Parameters
        xmlns:b=\"http://schemas.datacontract.org/2004/07/System.Collections.Generic\">
        ",
            "<a:KeyValuePairOfstringanyType>",
            "<b:key>EntityFilters</b:key>",
            "<b:value i:type=\"c:EntityFilters\""
        xmlns:c=\"http://schemas.microsoft.com/xrm/2011/Metadata\">
            + _xmlEncode(entityFiltersValue) + "</b:value>",

            "</a:KeyValuePairOfstringanyType>",
            "<a:KeyValuePairOfstringanyType>",
            "<b:key>MetadataId</b:key>",
            "<b:value i:type=\"ser:guid\""
        xmlns:ser=\"http://schemas.microsoft.com/2003/10/Serialization\">
            + _xmlEncode(MetadataId) + "</b:value>",

            "</a:KeyValuePairOfstringanyType>",
            "<a:KeyValuePairOfstringanyType>",
            "<b:key>RetrieveAsIfPublished</b:key>",
            "<b:value i:type=\"c:boolean\""
        xmlns:c=\"http://www.w3.org/2001/XMLSchema\">
            + _xmlEncode(RetrieveAsIfPublished.toString()) + "</b:value>",

            "</a:KeyValuePairOfstringanyType>",
            "<a:KeyValuePairOfstringanyType>",
            "<b:key>LogicalVarN</b:key>",
            entityLogicalVarNValueNode,
            "</a:KeyValuePairOfstringanyType>",

            "</a:Parameters>",

            "<a:RequestId i:nil=\"true\" />",

            "<a:RequestVarN>RetrieveEntity</a:RequestVarN>",

            "</request>",

            "</Execute>",

            "</soapenv:Body>",

            "</soapenv:Envelope>"].join("");
        var req = new XMLHttpRequest();
        // Changed by Neomy
        //req.open("POST", _getUrl() +
        "/XRMServices/2011/Organization.svc/web", true);
        req.open("POST", _getUrl() +
        "/XRMServices/2011/Organization.svc/web", false);
        try { req.responseType = 'msxml-document' } catch (e) { }
        req.setRequestHeader("Accept", "application/xml, text/xml,
/*");
        req.setRequestHeader("Content-Type", "text/xml; charset=utf-
8");
        req.setRequestHeader("SOAPAction",
        "http://schemas.microsoft.com/xrm/2011/Contracts/Services/IOrganizationService/
Execute");

        req.onreadystatechange = function () {
            if (req.readyState == 4 /* complete */) {
                req.onreadystatechange = null; //Addresses potential
memory leak issue with IE
                if (req.status == 200) {
                    var doc = req.responseXML;
                    try { _setSelectionVarNspaces(doc); } catch (e) { }
                    var a = _objectifyNode(_selectSingleNode(doc,
"/b:value"));

                    a._type = "EntityMetadata";
                    successCallBack(a, passThroughObject);
                }
            else {
                //Failure

```

```

                errorCallBack(_getError(req));
            }
        }

    };
    req.send(request);

};

function _getError(resp) {
    ///<summary>
    /// Private function that attempts to parse errors related to
    connectivity or WCF faults.
    ///</summary>
    ///<param VarN="resp" type="XMLHttpRequest">
    /// The XMLHttpRequest representing failed response.
    ///</param>

    //Error descriptions come from
    http://support.microsoft.com/kb/193625
    if (resp.status == 12029)
    { return new Error("The attempt to connect to the server
failed."); }
    if (resp.status == 12007)
    { return new Error("The server VarN could not be resolved."); }
    var faultXml = resp.responseXML;
    var errorMessage = "Unknown (unable to parse the fault)";
    if (typeof faultXml == "object") {

        var faultstring = null;
        var ErrorCode = null;

        var bodyNode = faultXml.firstChild.firstChild;

        //Retrieve the fault node
        for (var i = 0; i < bodyNode.childNodes.length; i++) {
            var node = bodyNode.childNodes[i];

            //NOTE: This comparison does not handle the case where
            the XML VarNspace changes
            if ("s:Fault" == node.nodeVarN) {
                for (var j = 0; j < node.childNodes.length; j++) {
                    var testNode = node.childNodes[j];
                    if ("faultstring" == testNode.nodeVarN) {
                        faultstring = _getNodeText(testNode);
                    }
                    if ("detail" == testNode.nodeVarN) {
                        for (var k = 0; k <
testNode.childNodes.length; k++) {
                            var orgServiceFault =
testNode.childNodes[k];
                            if ("OrganizationServiceFault" ==
orgServiceFault.nodeVarN) {
                                for (var l = 0; l <
orgServiceFault.childNodes.length; l++) {
                                    var ErrorCodeNode =
orgServiceFault.childNodes[l];
                                    if ("ErrorCode" ==
ErrorCodeNode.nodeVarN) {
                                        ErrorCode =
_getNodeText(ErrorCodeNode);
                                        break;
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}

```



```

        "OneToManyRelationships",
        "Privileges",
        "LocalizedLabels",
        "Options",
        "Targets"];
    for (var i = 0; i < _arrayElements.length; i++) {
        if (elementVarN == _arrayElements[i]) {
            return true;
        }
    }
    return false;
};

function _objectifyNode(node) {
    //Check for null
    if (node.attributes != null && node.attributes.length == 1) {
        if (node.attributes.getVarNdItem("i:nil") != null &&
node.attributes.getVarNdItem("i:nil").nodeValue == "true") {
            return null;
        }
    }

    //Check if it is a value
    if ((node.firstChild != null) && (node.firstChild.nodeType ==
3)) {
        var nodeVarN = _getNodeVarN(node);

        switch (nodeVarN) {
            //Integer Values
            case "ActivityTypeMask":
            case "ObjectTypeCode":
            case "ColumnNumber":
            case "DefaultFormValue":
            case ".MaxValue":
            case ".MinValue":
            case "MaxLength":
            case "Order":
            case "Precision":
            case "PrecisionSource":
            case "LanguageCode":
                return parseInt(node.firstChild.nodeValue, 10);
                // Boolean values
            case "AutoRouteToOwnerQueue":
            case "CanBeChanged":
            case "CanTriggerWorkflow":
            case "IsActivity":
            case "IsAIRUpdated":
            case "IsActivityParty":
            case "IsAvailableOffline":
            case "IsChildEntity":
            case "IsCustomEntity":
            case "IsCustomOptionSet":
            case "IsDocumentManagementEnabled":
            case "IsEnabledForCharts":
            case "IsGlobal":
            case "IsImportable":
            case "IsIntersect":
            case "IsManaged":
            case "IsReadingPaneEnabled":
            case "IsValidForAdvancedFind":
            case "CanBeSecuredForCreate":
            case "CanBeSecuredForRead":
            case "CanBeSecuredForUpdate":

```

```

        case "IsCustomAttribute";
        case "IsManaged";
        case "IsPrimaryId";
        case "IsPrimaryVarN";
        case "IsSecured";
        case "IsValidForCreate";
        case "IsValidForRead";
        case "IsValidForUpdate";
        case "IsCustomRelationship";
        case "CanBeBasic";
        case "CanBeDeep";
        case "CanBeGlobal";
        case "CanBeLocal";
            return (node.firstChild.nodeValue == "true") ? true
        : false;
            //OptionMetadata.Value and
BooleanManagedProperty.Value and AttributeRequiredLevelManagedProperty.Value
        case "Value";
            //BooleanManagedProperty.Value
            if ((node.firstChild.nodeValue == "true") ||
(node.firstChild.nodeValue == "false")) {
                return (node.firstChild.nodeValue == "true") ?
true : false;
            }
            //AttributeRequiredLevelManagedProperty.Value
            if (
(node.firstChild.nodeValue == "ApplicationRequired")
||

(node.firstChild.nodeValue == "None") ||
(node.firstChild.nodeValue == "Recommended") ||
(node.firstChild.nodeValue == "SystemRequired")
) {
                return node.firstChild.nodeValue;
            }
            var numberValue =
parseInt(node.firstChild.nodeValue, 10);
            if (isNaN(numberValue)) {
                //FormatVarN.Value
                return node.firstChild.nodeValue;
            }
            else {
                //OptionMetadata.Value
                return numberValue;
            }
            break;
            //String values
        default:
            return node.firstChild.nodeValue;
    }
}

//Check if it is a known array
if (_is_M_D(_getNodeVarN(node))) {
    var arrayValue = [];

    for (var i = 0; i < node.childNodes.length; i++) {
        var objectTypeVarN;
        if ((node.childNodes[i].attributes != null) &&
(node.childNodes[i].attributes.getVarNdItem("i:type") != null)) {
            objectTypeVarN =
node.childNodes[i].attributes.getVarNdItem("i:type").nodeValue.split(":")[1];

```

```

        }
        else {
            objectTypeVarN = _getNodeVarN(node.childNodes[i]);
        }

        var b = _objectifyNode(node.childNodes[i]);
        b._type = objectTypeVarN;
        arrayValue.push(b);
    }

    return arrayValue;
}

//Null entity description labels are returned as <label/> - not
using i:nil = true;
if (node.childNodes.length == 0) {
    return null;
}

//Otherwise return an object
var c = {};
if (node.attributes.getVarNdItem("i:type") != null) {
    c._type =
node.attributes.getVarNdItem("i:type").nodeValue.split(":")[1];
}
for (var i = 0; i < node.childNodes.length; i++) {
    if (node.childNodes[i].nodeType == 3) {
        c[_getNodeVarN(node.childNodes[i])] =
node.childNodes[i].nodeValue;
    }
    else {
        c[_getNodeVarN(node.childNodes[i])] =
_objectifyNode(node.childNodes[i]);
    }
}
return c;
};

function _selectNodes(node, XPathExpression) {
    if (typeof (node.selectNodes) != "undefined") {
        return node.selectNodes(XPathExpression);
    }
    else {
        var out_a = [];
        var XPathResults = node.evaluate(XPathExpression, node,
_NSResolver, XPathResult.ANY_TYPE, null);
        var result = XPathResults.iterateNext();
        while (result) {
            out_a.push(result);
            result = XPathResults.iterateNext();
        }
        return out_a;
    }
};

function _selectSingleNode(node, xpathExpr) {
    if (typeof (node.selectSingleNode) != "undefined") {
        return node.selectSingleNode(xpathExpr);
    }
}
```

```

        else {
            var xpe = new XPathEvaluator();
            var m_p_t_N = xpe.evaluate(xpathExpr, node, _NSResolver,
XPathResult.FIRST_ORDERED_NODE_TYPE, null);
            return (m_p_t_N != null) ? m_p_t_N.singleNodeValue : null;
        }
    };

    function _selectSingleNodeText(node, xpathExpr) {
        var x = _selectSingleNode(node, xpathExpr);
        if (_isNodeNull(x))
        { return null; }
        if (typeof (x.text) != "undefined") {
            return x.text;
        }
        else {
            return x.textContent;
        }
    };

    function _getNodeText(node) {
        if (typeof (node.text) != "undefined") {
            return node.text;
        }
        else {
            return node.textContent;
        }
    };

    function _isNodeNull(node) {
        if (node == null)
        { return true; }
        if ((node.attributes.getVarNdItem("i:nil") != null) &&
(node.attributes.getVarNdItem("i:nil").value == "true"))
        { return true; }
        return false;
    };
    function _getNodeVarN(node) {
        if (typeof (node.baseVarN) != "undefined") {
            return node.baseVarN;
        }
        else {
            return node.localVarN;
        }
    };

    function _setSelectionVarNspaces(doc) {
        var VarNspaces = [
            "xmlns:s='http://schemas.xmlsoap.org/soap/envelope/'",
            "xmlns:a='http://schemas.microsoft.com/xrm/2011/Contracts'",
            "xmlns:i='http://www.w3.org/2001/XMLSchema-instance'",

"xmlns:b='http://schemas.datacontract.org/2004/07/System.Collections.Generic'",
            "xmlns:c='http://schemas.microsoft.com/xrm/2011/Metadata'"
        ];
        doc.setProperty("SelectionVarNspaces", VarNspaces.join(" "));
    }

    function _NSResolver(prefix) {
        var ns = {
            "s": "http://schemas.xmlsoap.org/soap/envelope/",

```

```

        "a": "http://schemas.microsoft.com/xrm/2011/Contracts",
        "i": "http://www.w3.org/2001/XMLSchema-instance",
        "b":
    "http://schemas.datacontract.org/2004/07/System.Collections.Generic",
        "c": "http://schemas.microsoft.com/xrm/2011/Metadata"
    );
    return ns[prefix] || null;
};

function _xmlEncode(strInput) {
    var c;
    var XmlEncode = '';
    if (strInput == null) {
        return null;
    }
    if (strInput == '') {
        return '';
    }
    for (var cnt = 0; cnt < strInput.length; cnt++) {
        c = strInput.charCodeAt(cnt);
        if (((c > 96) && (c < 123)) ||
            ((c > 64) && (c < 91)) ||
            (c == 32) ||
            ((c > 47) && (c < 58)) ||
            (c == 46) ||
            (c == 44) ||
            (c == 45) ||
            (c == 95)) {
            XmlEncode = XmlEncode + String.fromCharCode(c);
        }
        else {
            XmlEncode = XmlEncode + '&#' + c + ';';
        }
    }
    return XmlEncode;
};

$( "#search_jqGrid" ).click();
$( "#searchmodfbox_jqGrid" ).width(500);
$( "#searchmodfbox_jqGrid" ).height(115);
$( "#fbox_jqGrid_reset" ).hide();
$( "#searchmodfbox_jqGrid" ).css('top', 5 + 'px');
$( "#searchmodfbox_jqGrid" ).css('left', 5 + 'px');
$( "#jqg1" ).width(100);
$(".ui-jqgrid-bdiv").height(100);
// $( "#del_jqGrid" ).hide();
$( "#add_jqGrid" ).hide();
$( "#edit_jqGrid" ).hide();
//alerthd_jqGrid
                //del_jqGrid

});

//on click city colum open new window with cities view
function openNewJQgrid() {
    var subStringToCheck = "input";
    var stringToCheck =
$("#jqGrid").getRowData($("#jqGrid").jqGrid('getGridParam',
'selrow')).FirstVarN;

```

```

        if (stringToCheck.indexOf(subStringToCheck) != -1) {
            var url = serverUrl +
"/WebResources/be_jqGrid_view_cities?preview=1";
            var myObject = new Object();
            myObject.Org = org;
            var result = showModalDialog(url, myObject,
"dialogHeight:300px;dialogWidth:500px;center:yes;
resizable:1;status:no;scroll:yes");
            if (result != undefined) {
                $("#" + jqGrid).jqGrid("setCell",
                $("#" + jqGrid).jqGrid('getGridParam', 'selrow'), "City", result retVal);
            }
        }
    }

function getAccountId() {
    //Get the any query string parameters and load them
    //into the vals array
    var vals = new Array();
    if (location.search != "") {
        vals = location.search.substr(1).split("&");
        for (var i in vals) {
            vals[i] = vals[i].replace(/\+/g, " ").split "=");
        }
        //look for the parameter VarNd 'data'
        var found = false;
        for (var i in vals) {
            if (vals[i][0].toLowerCase() == "data") {
                return vals[i][1];
            }
        }
    }
    //get contact search type (Patient or Representative) by query string
    parameter
    function getDataParam() {
        //Get the any query string parameters and load them
        //into the vals array
        var searchType = null;
        var vals = new Array();
        if (location.search != "") {
            vals = location.search.substr(1).split("&");
            for (var i in vals) {
                vals[i] = vals[i].replace(/\+/g, " ").split "=");
            }
            //look for the parameter VarNd 'data'
            var found = false;
            for (var i in vals) {
                if (vals[i][0].toLowerCase() == "data") {
                    searchType = vals[i][1];
                    found = true;
                    break;
                }
            }
        }
        return searchType;
        //    if (!found)
        //    { noParams(); }
    //}
    //else {
    //    noParams();
}

```

```

        //}
    }
    //get value from url
    GetValueFormUrl = function (paramter) {
        var vars = {};
        var result;
        if (window.location.search.length !== 0)
            window.location.search.replace(/[^?&]+(&[^=]+)=([^\&]*)/gi,
        function (m, key, value) {
            key = decodeURIComponent(key);
            if (typeof vars[key] === "undefined") { vars[key] =
decodeURIComponent(value); }
            else { vars[key] = [].concat(vars[key],
decodeURIComponent(value)); }
        });
        result = vars[paramter];
        return result;
    }
    //remove brackets
    removeBrackets = function (str) {
        var guid;
        if (str != undefined) {
            guid = str.replace('{', '').replace('}', '');
        }
        return guid;
    }
</script>

</body>
</html>

```

be_searchRepresentative

<html>

```
<head>
    <meta http-equiv="x-ua-compatible" content="IE=edge">
    <meta charset="utf-8">
    <!-- The jQuery library is a prerequisite for all jqSuite products -->
    <script src="homL_jquery" type="text/javascript"></script>
    <script src="homL_json" type="text/javascript"></script>
    <!-- This is the Javascript file of jqGrid -->
    <!-- <script type="text/javascript" src="homL_jqgrid"></script>-->
    <script src="homL_jquery.jqGridTest" type="text/javascript"></script>
    <!-- This is the localization file of the grid controlling messages,
labels, etc.
    <!-- We support more than 40 localizations -->
    <!-- <script type="text/javascript" src="homL_locale_en"></script>-->
    <script src="homL_grid.localeenTest" type="text/javascript"></script>
    <!-- <script type="text/javascript" src="homL_grid.localeheTest"></script>-->
->
    <script src="homL_jquery.ui" type="text/javascript"></script>
    <script src="SDK.MetaData" type="text/javascript"></script>
    <!-- A link to a jQuery UI ThemeRoller theme, more than 22 built-in and
many more custom -->
    <link href="homL_jquery_ui" rel="stylesheet" type="text/css"
media="screen">
    <!-- The link to the CSS that the grid needs -->
    <link href="homL_ui_jqgrid" rel="stylesheet" type="text/css"
media="screen">
<title>Search Contacts</title>

</head>
[REDACTED]
```

```
[REDACTED]
[REDACTED]
[REDACTED]
[REDACTED]

var el = document.getElementById('lnkSearch');
el.onclick = openSearchWindow;
var PR = window.location.PR;
var hostVarN = window.location.hostVarN;
var path = window.location.pathVarN.split("/");
org = path[1];
serverUrl = PR + "//" + hostVarN + "/" + org;

function openSearchWindow() {
    var recordId = window.parent.Xrm.Page.data.entity.getId();
    if (recordId != "") {
        var searchType = "?id=" + recordId + "&data=representative";
        var retObj = window.showModalDialog(serverUrl +
'/WebResources/be_search_contacts' + searchType, null,
"dialogHeight:200px;dialogWidth:1200px;center:yes;
resizable:1;status:no;scroll:yes");
        var representative = retrieveRepresentative(recordId);
        if (representative != null) {

window.parent.Xrm.Page.getAttribute("be_representativeVarN").setValue(representative.be_representativeVarN);

window.parent.Xrm.Page.getAttribute("be_representativephone").setValue(representative.be_representativephone);
```

```

window.parent.Xrm.Page.getAttribute("emailaddress").setValue(representative.emai
iladdress);

window.parent.Xrm.Page.getAttribute("be_representativeidnumber").setValue(repre
sentative.be_representativeidnumber);

window.parent.Xrm.Page.getAttribute("be_patientfullVarN").fireOnChange();

window.parent.Xrm.Page.getAttribute("be_dateofbirth").fireOnChange();
}

}

function retrieveRepresentative(recordId) {
    var representative;
    var req = new XMLHttpRequest();
    req.open("GET", serverUrl +
"/api/data/v8.0/be_benefits_exhaustion_incidents(" + removeBrackets(recordId) +
")?$select=be_representativeVarN,be_representativephone,emailaddress,be_repre
sentativeidnumber,be_representativeothertype", false);
    req.setRequestHeader("OData-MaxVersion", "4.0");
    req.setRequestHeader("OData-Version", "4.0");
    req.setRequestHeader("Accept", "application/json");
    req.setRequestHeader("Content-Type", "application/json;
charset=utf-8");
    req.setRequestHeader("Prefer", "odata.include-annotations='*\"");
    req.onreadystatechange = function () {
        if (this.readyState === 4) {
            req.onreadystatechange = null;
            if (this.status === 200) {
                representative = JSON.parse(this.response);
                // return result;
            } else {
                Xrm.Utility.alertDialog(this.statusText);
            }
        }
    };
    req.send();
    return representative;
}

//remove brackets
removeBrackets = function (str) {
    var guid;

    if (str != undefined) {
        guid = str.replace('{', '').replace('}', '');
    }
    return guid;
}
</script>

</body>
</html>

```

connectEmailToIncident

```
<html>
<head>
    <meta charset="utf-8">
    <script>
        function ConnectToIncident()
        {
            debugger;
            var xrmPage = window.parent.Xrm.Page;
            var emailId =
xrmPage.data.entity.getId().replace('{', '').replace('}', '');
            var regardingobjectid = xrmPage.getAttribute("regardingobjectid");
            var incident = txtIncident.value;
            var clientUrl = xrmPage.context.getClientUrl();
```

```
        }
        function retrieveIncident(clientUrl,incident)
        {
            var incidentId;
            var req = new XMLHttpRequest();
            req.open("GET",clientUrl + "/api/data/v8.0/incidents?filter eq '" + incident + "'&$select=incidentid",true);
            req.onreadystatechange = function() {
                if (this.readyState === 4) {
                    req.onreadystatechange = null;
                    if (this.status === 200) {
                        var results = JSON.parse(this.response);
                        if (results.value.length != 0) {
                            for (var i = 0; i < results.value.length; i++) {
                                incidentId =
```

```
[REDACTED]  
}  
  
function UpdateEmail(xrmPage, incidentid, emailid) {  
  
    if (xrmPage != null && incidentid != null && emailid != null) {  
        var entity = {};  
  
entity["regardingobjectid_be_benefits_exhaustion_incident@odata.bind"] =  
"/be_benefits_exhaustion_incidents(" + incidentid + ")";  
  
        var req = new XMLHttpRequest();  
        req.open("PATCH", xrmPage.context.getClientUrl() +  
"/api/data/v8.0/emails(" + emailid + ")", false);  
        req.setRequestHeader("OData-MaxVersion", "4.0");  
        req.setRequestHeader("OData-Version", "4.0");  
        req.setRequestHeader("Accept", "application/json");  
        req.setRequestHeader("Content-Type", "application/json;  
charset=utf-8");  
        req.setRequestHeader("Prefer", "odata.include-  
annotations='*\"");  
[REDACTED]  
[REDACTED]  
[REDACTED]  
[REDACTED]  
[REDACTED]  
[REDACTED]  
        req.send(JSON.stringify(entity));  
    }  
}  
}</script>  
<style>  
    .button {  
        margin-right: 8px;  
        height: 24px;  
        font-family: Segoe UI,Tahoma,Arial;  
        border: 1px solid #C6C6C6;  
        background-image: none;  
        margin-top: 10px;  
        width: auto !important;  
        min-width: 80px;  
        white-space: nowrap;  
        color: #444444;  
        cursor: pointer;  
        background: transparent;  
        background-color: #FFFFFF;  
        font: Tahoma;  
        width: 93px;  
        height: 23px;  
    }  
  
    .button:hover {  
        background-color: rgba(214, 234, 248, 1);  
    }  
</style>  
<title>הנעה לשינוי</title>  
</head>
```

```
<body style="direction: rtl; overflow-wrap: break-word;">
  <strong>
    <font face="Tahoma" size="1">
      <font size="1">&nbsp;</font><input id="txtIncident" style="width: 91px; height: 21px;" type="text" size="1">&nbsp;
      &nbsp; <button class="button" id="bt_ok" onclick="ConnectToIncident()">אישור</button>
      &nbsp;&nbsp;&nbsp;
      &nbsp;
    </font>
  </strong>

</body>
</html>
```

be_search_contact_dashboard

```
<html>

<head>
    <meta http-equiv="x-ua-compatible" content="IE=9 ie=edge">
    <meta charset="utf-8">
    <!-- The jQuery library is a prerequisite for all jqSuite products -->
    <script src="homL_jquery" type="text/javascript"></script>
    <script src="homL_json" type="text/javascript"></script>
    <!-- This is the Javascript file of jqGrid -->
    <!-- <script type="text/javascript" src="homL_jqgrid"></script>-->
    <script src="homL_jquery.jqGridTest" type="text/javascript"></script>
    <!-- This is the localization file of the grid controlling messages,
labels, etc.
    <!-- We support more than 40 localizations -->
    <!-- <script type="text/javascript" src="homL_locale_en"></script>-->
    <script src="homL_grid.localeenTest" type="text/javascript"></script>
    <!-- <script type="text/javascript" src="homL_grid.localeheTest"></script>-->
    <script src="homL_jquery.ui" type="text/javascript"></script>
    <!-- A link to a jQuery UI ThemeRoller theme, more than 22 built-in and
many more custom -->
    <link href="homL_jquery_ui" rel="stylesheet" type="text/css"
media="screen">
    <!-- The link to the CSS that the grid needs -->
    <link href="homL_ui_jqgrid" rel="stylesheet" type="text/css"
media="screen">
    <title>jqGrid Loading Data - JSON Live</title>

</head>
<body style="direction: rtl;">
    <style type="text/css">
        /* set the size of the datepicker search control for Order Date*/
        #ui-datepicker-div {
            font-size: 11px;
        }
        /* set the size of the autocomplete search control*/
        .ui-menu-item {
            font-size: 11px;
        }

        .ui-autocomplete {
            font-size: 11px;
            position: absolute;
            cursor: default;
            z-index: 5000 !important;
        }
    </style>

    <div align="right">
        <table id="jqGrid"></table>
    </div>
    <div align="center" id="jqGridPager"></div>

    <script type="text/javascript">
        var serverUrl;
        var odataPath11;
        var odataPath16;
        var org;
        var isValid = true;
```

