

```

<!DOCTYPE html><html xmlns="http://www.w3.org/1999/xhtml"><head>
<title>Opportunity Products</title>
<script src="/WebResources/ClientGlobalContext.js.aspx"></script>
<script type="text/javascript" src="new_new_jquery"></script>
<script type="text/javascript" src="new_new_knockout_220"></script>
<script type="text/javascript" src="new_SDK_REST.js"></script>
<script type="text/javascript" src="new_SDK.Metadata"></script>
<link rel="stylesheet" type="text/css" href="new_new_normalize"/>
<style type="text/css">body { font-family: Segoe UI,Tahoma,Arial; font-size: 14px;
} .liveExample { padding: 1em; border: 1px solid #CCC; max-width: 100%; text-
align: left; } .liveExample input { font-family: Segoe UI,Tahoma,Arial; }
.liveExample b { font-weight: bold; } .liveExample p { margin-top: 0.9em; margin-
bottom: 0.9em; } .liveExample select[multiple] { width: 100%; height: 8em; }
.liveExample h2 { margin-top: 0.4em; font-weight: bold; font-size: 1.2em; }
.liveExample table, .liveExample td, .liveExample th { padding: 0.2em; border-
width: 0; } .liveExample td input { width: 13em; } tr { vertical-align: top; }
.liveExample input.error { border: 1px solid red; background-color: #FDC; }
.liveExample label.error { display: block; color: Red; font-size: 0.8em; }
.liveExample th { font-weight: bold; } li { list-style-type: disc; margin-left:
20px; } </style>
<meta charset="utf-8"/>
</head><body>
<div class="liveExample">
  <table data-bind="visible: true">
    <thead>
      <tr>
        <th>Product</th>
        <th>UOM</th>
        <th>Projected Schedule</th>
        <th>Quantity</th>
      </tr>
    </thead>
    <tbody data-bind="foreach: oproducts">
      <tr>
        <td><span data-bind="text: ProductId.Name"></span><a href="#"
data-bind="
        click: $root.lookupProduct">Lookup</a></td>
        <td><span data-bind="text: UoId.Name"></span><a href="#" data-
bind="
        click: $root.lookupUOM ">Lookup</a></td>
        <td><select data-bind="options: $root.availableSchedule,
optionsText: 'ProjectedScheduleText', value: new_projectedschedule.Value,
optionsValue: 'ProjectedScheduleValue', optionsCaption:
'Choose...'"></select></td>
        <td><input data-bind="value: Quantity"/></td>
        <td><a href="#" data-bind="click:
$root.removeOppProd">Delete</a></td>
      </tr>
    </tbody>
  </table>
  <button data-bind="click: addOppProd">Add Opportunity Product</button>
  <button data-bind="click: save">Save</button>
</div>
<script type="text/javascript">
var OpportunityProductModel = function () {
  var self = this;
  self.oproducts = ko.observableArray([]);
  self.availableSchedule = ko.observableArray([]);
  self.RetrieveOptionsetData = function () {
    // Entity schema name
    var entityLogicalName = "opportunityproduct";
    // option set schema name

```

```

var RetrieveAttributeName = "new_projectedschedule";
// Calling Metadata service to get Optionset Label
SDK.Metadata.RetrieveEntity(SDK.Metadata.EntityFilters.Attributes,
entityLogicalName, null, false,
function (entityMetadata) {
    //
    // result
    //
    for (var i = 0; i < entityMetadata.Attributes.length; i++) {
        var AttributeMetadata = entityMetadata.Attributes[i];
        if (AttributeMetadata.SchemaName.toLowerCase() ==
RetrieveAttributeName.toLowerCase()) {
            for (var o = 0; o <
AttributeMetadata.OptionSet.Options.length; o++) {
                var option = AttributeMetadata.OptionSet.Options[o];
                self.availableSchedule.push(new
ProjectedSchedule(option.Label.UserLocalizedLabel.Label, option.Value)); // Adds
another option
            }
        }
    }

    //
    // fetch opportunity product data
    //
    var guid = window.parent.Xrm.Page.data.entity.getId();

    if (guid != null) {
        if (guid.indexOf("{") == 0) {
            guid = guid.substr(1, (guid.length - 2)).toLowerCase();

            // Load initial state from server, convert it to Task
instances, then populate self.tasks
            var filter =
"?$select=OpportunityId,OpportunityProductId,Quantity,ProductId,IsPriceOverridden,
UoMId,new_projectedschedule,product_opportunities/PriceLevelId&$expand=product_opp
ortunities&$filter=OpportunityId/Id eq guid'" + guid + "'";

            SDK.REST.retrieveMultipleRecords(
                "OpportunityProduct",
                filter,
                function (data) {

                    var mappedTasks = new Array();

                    if (data) {

                        console.log("JSON received: " +
JSON.stringify(data));

                        var mappedTasks = $.map(data, function (item) {
                            console.log("item: " + JSON.stringify(item));
                            return new OppProduct(item)
                        });

                        self.opproducts(mappedTasks);
                    }
                }, errorHandler,
                function () {
                    //OnComplete handler

```

```

        }
    );
}
}, self.errorRetrieveEntity,
    false);

};

self.errorRetrieveEntity = function (error) {
    ///
    /// Displays the error returned from SDK.Metadata.RetrieveEntity if it
fails.
    ///
    alert("Error!" + error);
};

//
// addOppProd - ADD NEW
// insert new opportunity product
//
self.addOppProd = function () {

    var surl = "";
    var dialogArgs = {};
    var productid;

    surl = "../.." + getQuerystring("Default") +
    "_controls/lookup/lookupinfo.aspx?AllowFilterOff=1&DefaultType=1024&DefaultViewId=
%7bA2D479C5-53E3-4C69-ADDD-
802327E67A0D%7d&DisableQuickFind=0&DisableViewPicker=0&LookupStyle=single&ShowNewB
utton=1&ShowPropButton=1&browse=0&objecttypes=1024";

    var oppprod = new OppProductConstructor();
    var lookup = new Array();

    var oLookupItems = showModalDialog(surl, dialogArgs,
'dialogWidth:600px;dialogHeight:350px;resizable:yes;status:yes');
    if (oLookupItems != null) {
        var obj = jQuery.parseJSON(oLookupItems);
        console.log("obj: " + JSON.stringify(obj));

        if (obj.items.length > 0) {

            oppprod.ProductId = new ProductIdFromLookup(obj.items[0]);
            productid = obj.items[0].id;

            var prodguid = productid;

            if (prodguid != null) {
                if (prodguid.indexOf("{") == 0) {
                    prodguid = prodguid.substr(1, (prodguid.length -
2)).toLowerCase();
                }
            }

            // set opportunity Id
            var guid = window.parent.Xrm.Page.data.entity.getId();

            if (guid != null) {

```

```

        if (guid.indexOf("{") == 0) {
            guid = guid.substr(1, (guid.length - 2)).toLowerCase();
        }
        oppprod.OpportunityId = { Id: guid, LogicalName: "opportunity"
};
    }

    // get price list from opportunity
    var filter =
"?$select=Name,OpportunityId,PriceLevelId&$filter=OpportunityId eq guid'" + guid +
""";

    SDK.REST.retrieveMultipleRecords("Opportunity",
    filter,
    function (data) {

        if (data) {

            console.log("JSON received: " + JSON.stringify(data[0]));

            var pricelevelid = data[0].PriceLevelId.Id;

            // get unit and pricelists from from product
            var filter =
"?$select=DefaultUoMId,Name,ProductId,price_level_products/PriceLevelId&$expand=pr
ice_level_products&$filter=ProductId eq guid'" + prodguid + """;

            SDK.REST.retrieveMultipleRecords("Product",
                filter,
            function (data) {

                var haspricelevel = false;

                if (data) {
                    console.log("JSON received: " +
JSON.stringify(data[0]));
                    oppprod.UoMId = new UoMId(data[0].DefaultUoMId);

                    // match pricelist to opportunity pricelist
                    for (i = 0; i < data.length; i++) {
                        if (data[i].price_level_products != null &&
data[i].price_level_products.PriceLevelId == pricelevelid) {
                            haspricelevel = true;
                            break;
                        }
                    }

                    if (!haspricelevel) {
                        alert("The selected product does not have
price for the price list specified in the opportunity.");
                        return;
                    }

                    // create Opp Product
                    //
                    SDK.REST.createRecord(
                    ko.toJS(oppprod),
                    "OpportunityProduct",
                    function (result) {
                        var oppprodId = result.OpportunityProductId;

```

```

ko.observable(oppprodId);
                                oppprod.OpportunityProductId =
                                self.opproducts.push(oppprod);

                                },
                                function (error) {
                                    alert(error.message);
                                }
                                );
                                }
                                }, errorHandler,
                                function () {
                                    //OnComplete handler
                                }
                                );
                                }

                                }, errorHandler,
                                function () {
                                    //OnComplete handler
                                }
                                );
                                }
                                }

};
//
// END - ADD

//
// removeOppProd - DELETE
//
self.removeOppProd = function (item) {

    SDK.REST.deleteRecord(
        item.OpportunityProductId(),
        "OpportunityProduct",
        function () {
            self.opproducts.remove(item);
        },
        function (error) {
            alert("record not delete" + error);
        }
    );

};

//
// lookupProduct
//
self.lookupProduct = function (newitem) {

    var surl = "";
    var dialogArgs = {};

    surl = "../.." + getQueryString("Default") +
    "_controls/lookup/lookupinfo.aspx?AllowFilterOff=1&DefaultType=1024&DefaultViewId=
    %7bA2D479C5-53E3-4C69-ADDD-

```

```
802327E67A0D%7d&DisableQuickFind=0&DisableViewPicker=0&LookupStyle=single&ShowNewButton=1&ShowPropButton=1&browse=0&objecttypes=1024";
```

```
    var lookup = new Array();

    var oLookupItems = showModalDialog(surl, dialogArgs,
'dialogWidth:600px;dialogHeight:350px;resizable:yes;status:yes');
    if (oLookupItems != null) {
        var obj = jQuery.parseJSON(oLookupItems);

        if (obj.items.length > 0) {
            var guid = obj.items[0].id;
            if (guid != null) {
                if (guid.indexOf("{") == 0) {
                    guid = guid.substr(1, (guid.length - 2)).toLowerCase();

                    // set updated product id
                    newitem.ProductId.Id(guid);
                }
            }

            // set product name
            newitem.ProductId.Name(obj.items[0].name);
        }
    }
};

//
// lookup for UOM
//
self.lookupUOM = function (newitem) {

    var surl = "";
    var dialogArgs = {};

    var guid1 = newitem.OpportunityId.Id();
    var prodguid = newitem.ProductId.Id();
    surl = "../.." + getQuerystring("Default") +
"_controls/lookup/lookupinfo.aspx?AllowFilterOff=1&DefaultType=1055&DefaultViewId=
%7b26FF4574-E5B3-468D-B879-
8BB27225D70B%7d&DisableQuickFind=0&DisableViewPicker=0&LookupStyle=single&ObjectID
=%7b" + guid1 + "%7d&ObjectTypeCode=3&ProductId=%7b" + prodguid +
"%7d&ShowNewButton=0&ShowPropButton=1&browse=true&currentObjectType=1083&currentid
=%7bEDCEE641-9BDC-E111-B4D8-00059A3C7800%7d&objecttypes=1055";

    var lookup = new Array();

    var oLookupItems = showModalDialog(surl, dialogArgs,
'dialogWidth:600px;dialogHeight:350px;resizable:yes;status:yes');
    if (oLookupItems != null) {
        var obj = jQuery.parseJSON(oLookupItems);

        if (obj.items.length > 0) {
            var guid = obj.items[0].id;
            if (guid != null) {
                if (guid.indexOf("{") == 0) {
                    guid = guid.substr(1, (guid.length - 2)).toLowerCase();
                    newitem.UoMId.Id(guid);
                }
            }
        }
    }
};
```

```

        }
        newitem.UoMId.Name(obj.items[0].name);
    }
};

//
// save - SAVE
//
self.save = function () {

    var successcount = 0;
    for (i = 0; i < self.oproducts().length; i++) {
        if (self.oproducts()[i].Quantity == null) {
            alert('quantity is required');
            break;
        }

        if (self.oproducts()[i].ProductId == null) {
            alert('product is required');
            break;
        }

        if (self.oproducts()[i].UoMId == null) {
            alert('unit is required');
            break;
        }

        SDK.REST.updateRecord(
            self.oproducts()[i].OpportunityProductId(),
            ko.toJS(self.oproducts()[i]),
            "OpportunityProduct",
            function () {
                successcount++;
                if (self.oproducts().length == successcount) {
                    alert("success count: " + successcount);
                }
            }
        );

        parent.window.Xrm.Page.data.entity.attributes.get("budgetamount").setValue(successcount);

        parent.window.Xrm.Page.data.entity.save();
    },
    function (error) {
        alert("error updating: " + error.message);
    }
);

};

//
// START - main code
//
self.RetrieveOptionsetData();
};

```

```

function __metadataConstructor1() {
    this.uri = "";
    this.type = "";
};

function __metadataConstructor2(uri, type) {
    if (uri != null) {
        this.uri = ko.observable(uri);
    }
    this.type = ko.observable(type);
};

function OpportunityIdConstructor() {
    this.Id = "";
    this.LogicalName = "opportunity";
    this.Name = "";
    this.__metadata = new __metadataConstructor2(null,
"Microsoft.Crm.Sdk.Data.Services.EntityReference");
};

function UoMidConstructor() {
    this.Id = "";
    this.LogicalName = "uom";
    this.Name = "";
    this.__metadata = new __metadataConstructor2(null,
"Microsoft.Crm.Sdk.Data.Services.EntityReference");
};

function new_projectedscheduleConstructor() {
    this.Value = null;
    this.__metadata = new __metadataConstructor2(null,
"Microsoft.Crm.Sdk.Data.Services.OptionSetValue");
};

function ProductIdConstructor() {
    this.Id = "";
    this.LogicalName = "product";
    this.Name = "";
    this.__metadata = new __metadataConstructor2(null,
"Microsoft.Crm.Sdk.Data.Services.EntityReference");
};

function PriceLevelIdConstructor() {
    this.Id = "";
    this.LogicalName = "pricelevel";
    this.Name = "";
    this.__metadata = new __metadataConstructor2(null,
"Microsoft.Crm.Sdk.Data.Services.EntityReference");
};

function OppProductConstructor() {

    this.Quantity = "1.00";
    this.OpportunityId = new OpportunityIdConstructor();
    this.ProductId = new ProductIdConstructor();
    this.UoMid = new UoMidConstructor();
    this.new_projectedschedule = new new_projectedscheduleConstructor();
    this.IsPriceOverridden = false;
};

```



```

};

function __metadata(data) {

    if (data.uri != null) {
        this.uri = ko.observable(data.uri);
    }
    this.type = ko.observable(data.type);
};

function OpportunityId(data) {
    this.Id = ko.observable(data.Id);
    this.LogicalName = ko.observable(data.LogicalName);
    this.Name = ko.observable(data.Name);
    this.__metadata = new __metadata(data.__metadata);
};

function ProductId(data) {
    this.Id = ko.observable(data.Id);
    this.LogicalName = ko.observable(data.LogicalName);
    this.Name = ko.observable(data.Name);
    this.__metadata = new __metadata(data.__metadata);
};

function ProductIdFromLookup(data) {

    this.Id = ko.observable(data.id);
    this.LogicalName = ko.observable("product");
    this.Name = ko.observable(data.name);
    this.__metadata = new __metadataConstructor2(null,
"Microsoft.Crm.Sdk.Data.Services.EntityReference");
};

function UoMId(data) {
    this.Id = ko.observable(data.Id);
    this.LogicalName = ko.observable(data.LogicalName);
    this.Name = ko.observable(data.Name);
    this.__metadata = new __metadata(data.__metadata);
};

function new_projectedschedule(data) {
    this.Value = ko.observable(data.Value);
    this.__metadata = new __metadata(data.__metadata);
};

var ProjectedSchedule = function (name, population) {
    this.ProjectedScheduleText = name;
    this.ProjectedScheduleValue = population;
};

function PriceLevelId(data) {
    this.Id = data.Id;
    this.LogicalName = data.LogicalName;
    this.Name = data.Name;
    this.__metadata = new __metadata(data.__metadata);
};

function OppProduct(data) {
    this.__metadata = new __metadata(data.__metadata);
};

```

```

        this.Quantity = ko.observable(data.Quantity);
        this.OpportunityProductId = ko.observable(data.OpportunityProductId);
        this.OpportunityId = new OpportunityId(data.OpportunityId);
        this.ProductId = new ProductId(data.ProductId);
        this.UoMId = new UoMId(data.UoMId);
        this.new_projectedschedule = new
new_projectedschedule(data.new_projectedschedule);
        this.IsPriceOverridden = data.IsPriceOverridden;

    };

    function getQuerystring(key, default_) {
        if (default_ == null)
            default_ = "";
        key = key.replace(/[\[\]\/, "\\\\[\"].replace(/[\\]/, "\\");
        var regex = new RegExp("[\\?&]" + key + "=(^[^#]*)");
        var qs = regex.exec(window.location.href);
        if (qs == null)
            return default_;
        else
            return qs[1];
    }

    function errorHandler(error) {
        alert(error.message);
    }

    function waitForScriptsAccount(ObjectNames, callback) {

        var hasLoaded = false;

        function checkScripts() {

            var allLoaded = true;

            for (var i = 0; i < ObjectNames.length; i++) {

                var hasLoaded = true;
                var ObjectName = ObjectNames[i];
                var Script = "typeof " + ObjectName + " == 'undefined'";

                hasLoaded = !eval(Script);
                allLoaded = allLoaded && hasLoaded;

                if (!allLoaded) { setTimeout(checkScripts, 10); break; }

            }
            if (allLoaded) { callback(); }
        }
        setTimeout(checkScripts, 0);
    }

    var _f_ = function () { };
    window.console = window.console || { log: _f_, error: _f_, info: _f_, debug:
_f_, warn: _f_, trace: _f_, dir: _f_, dirxml: _f_, group: _f_, groupEnd: _f_,
time: _f_, timeEnd: _f_, assert: _f_, profile: _f_ };

    waitForScriptsAccount(["SDK.REST", "jQuery"],
        function () {

```

```
        var pageViewModel = new OpportunityProductModel();
        ko.applyBindings(pageViewModel);
    });
</script>
</body></html>
```