**Set Lookup**

var lookupValue = new Array();

lookupValue[0] = new Object();

lookupValue[0].id = "727504ed-64c5-4bc8-ac22-0a3071c427e3"; // GUID of the lookup id

lookupValue[0].name = "contact name"; // Name of the lookup

lookupValue[0].entityType = "contact"; //Entity Type of the lookup entity

Xrm.Page.getAttribute("FieldName").setValue(lookupValue); // You need to replace the lookup field Name..

**To Get Lookup Value**

**function getlookupvalue(executioncontext)**

**{**

var formcontext=executioncontext.getformcontext();

var GetLookupid=formcontext.getAttribute("new").getValue()[0].id; //to get lookup id

var GetLookupName=formcontext.getAttribute("new").getValue()[0].name; //to get lookup name

var getlogicalname=formcontext.getAttribute("new").getValue()[0].entitytype; //to get lookup logicalname

**}**

**Get Text Value**

**function gettext(executioncontext)**

**{**

var formcontext=executioncontext.getformcontext();

var textvalue=formcontext.getAttribute("new").getValue();

**}**

**Get Optionsetvalue and text**

**function getOptionset(executioncontext)**

**{**

var formcontext=executioncontext.getformcontext();

var optionsetvalue=formcontext.getAttribute("new").getValue();

var optionsettext=formcontext.getAttribute("new").getText();

**}**

**Get Date File**

**function getDate(executioncontext)**

**{**

var formcontext=executioncontext.getformcontext();

var  getdatevalue=formcontext.getAttribute("new").getValue();

**}**

**To Set Text Field Value**

**function settextfield(executioncontext)**

**{**

var formcontext=executioncontext.getformcontext();

var  settextvalue=formcontext.getAttribute("new").setValue("text");

**}**

**To Set Two Way Option**

**function settwoway(executioncontext)**

**{**

var formcontext=executioncontext.getformcontext();

var  settwoway=formcontext.getAttribute("new").setValue(true);

var  settwoway1=formcontext.getAttribute("new").setValue(false);

**}**

**To Set Business Required**

**function setbussinessrequirment(executioncontext)**

**{**

var formcontext=executioncontext.getformcontext();

var  setbussinessrequired=formcontext.getAttribute("new").setRequiredlevel("required");

var  setbussinessrequired1=formcontext.getAttribute("new").setRequiredlevel("recommended");

var  setbussinessrequired2=formcontext.getAttribute("new").setRequiredlevel("none");

**}**

**To Set field as Read Only**

**function setfieldreadonly(executioncontext)**

**{**

var formcontext=executioncontext.getformcontext();

var  setfieldreadonly1=formcontext.getAttribute("new").setDisabled(true);

var  setfieldreadonly2=formcontext.getAttribute("new").setDisabled(false);

**}**

**Custom Filter for Lookup**

**function setfieldreadonly(executioncontext)**

**{**

var formcontext=executioncontext.getformcontext();

var filter="<fetch mapping='logical'>

   <entity name='account'>

      <attribute name='accountid'/>

      <attribute name='name'/>

</entity>

</fetch>";

var lookup=formcontext.getAttribute("new");

lookup.addpresearch(function()

{

lookup.addcustomfilter(filter);

});

}

**Show Field**

**function showfield(executioncontext)**

**{**

var formcontext=executioncontext.getformcontext();

var  showfield=formcontext.getAttribute("new").setVisible(true);

}

**Hide Field**

**function showfield(executioncontext)**

**{**

var formcontext=executioncontext.getformcontext();

var  hidefield=formcontext.getAttribute("new").setVisible(false);

}

**Show Organization or Hide Organization**

**function showorgorhide(executioncontext)**

**{**

var formcontext=executioncontext.getformcontext();

var  orgshow=formcontext.ui.navigation.items.get("org").setvisible(true); //show org

var  orghide=formcontext.ui.navigation.items.get("org").setvisible(false);//hide org

}

**Set Form Notification**

Parameter

| **PARAMETER** | | | |
| --- | --- | --- | --- |
| **Name** | **Type** | **Required** | **Description** |
| message | String | Yes | The text of the message. |
| level | String | Yes | The level of the message, which defines how the message will be displayed. Specify one of the following values: ERROR : Notification will use the system error icon. WARNING : Notification will use the system warning icon. INFO : Notification will use the system info icon. |
| uniqueId | String | Yes | A unique identifier for the message that can be used later with clearFormNotification to remove the notification. |

**function setnotification(executioncontext)**

{

var formcontext=executioncontext.getformcontext();

var  errorshow=formContext.ui.setFormNotification("errormessage", "ERROR","error12");

var  warningshow=formContext.ui.setFormNotification("warningmessage", "WARNING", "warning12");

var  infoshow=formContext.ui.setFormNotification("infomessage", "INFO", "info12");

}

**Clear Form Notification**

**function clearnotification(executioncontext)**

{

var formcontext=executioncontext.getformcontext();

var  errorshow=formContext.ui.clearFormNotification("error12");

var  warningshow=formContext.ui.clearFormNotification("warning12");

var  infoshow=formContext.ui.clearFormNotification("info12");

}

**To Show or Hide Tab**

**function showhidetab(executioncontext)**

{

var formcontext=executioncontext.getformcontext();

var  show=formContext.ui.tabs.get("new").setVisible(true); //to show tab

var  hide=formContext.ui.tabs.get("new").setVisible(false); //to hide tab

}

**To Show or Hide Section**

function showhidesections(executionContext) {

  //show section

  var tabObj = formContext.ui.tabs.get("tabname"); //the tab where your section is

var sectionObj = tabObj.sections.get("sectionname"); // section name

sectionObj.setVisible(true);

 //hide section

 var tabObj = formContext.ui.tabs.get("tabname"); //the tab where your section is

var sectionObj = tabObj.sections.get("sectionname"); // section name

sectionObj.setVisible(false);

}

Get Current Record Guid

**function currentrecordguid(executioncontext)**

{

var formcontext=executioncontext.getformcontext();

var  currentguid=formContext.data.entity.getId();

}

Get EntityName

**function EntityName(executioncontext)**

{

var formcontext=executioncontext.getformcontext();

var  entityname=formContext.data.entity.getEntityName();

}

GetDataXML

Returns a string representing the XML that will be sent to the server when the record is saved. Only data in fields that have changed are set to the server.

**function GetDataXML(executioncontext)**

{

var formcontext=executioncontext.getformcontext();

var  getdataxml=formContext.data.entity.getDataXml();

}

**Return Value**

**Type**: String.

**Description**: In this example, the following three fields for an account record were updated: name, accountnumber, telephone2.

"<account><name>Contoso</name><accountnumber>55555</accountnumber><telephone2>425 555-1234</telephone2></account>"

addOnSave

Adds a function to be called when the OnSave event is triggered. The event occurs before the save occurs, giving the handler an option to cancel the save operation.

**function addonsave(executioncontext)**

{

var formcontext=executioncontext.getformcontext();

var  addonsave=formContext.data.entity.addOnSave(GetDataXML(formcontext))

}

removeOnSave

Removes a function to be called when the record is saved.

**function removeonsave(executioncontext)**

{

var formcontext=executioncontext.getformcontext();

var  addonsave=formContext.data.entity.removeOnSave(GetDataXML)

}

save

Saves the record asynchronously with the option to set callback functions to be executed after the save operation is completed.

**function save(executioncontext)**

{

var formcontext=executioncontext.getformcontext();

var  save=formContext.data.save().then(function () {

    // success

{ saveMode: 2}

}, function (error) {

    // failure

    // you get more information about why saving data failed in error argument:

    var errorCode = error.errorCode;

    var errorMessage= error.message;

});

}