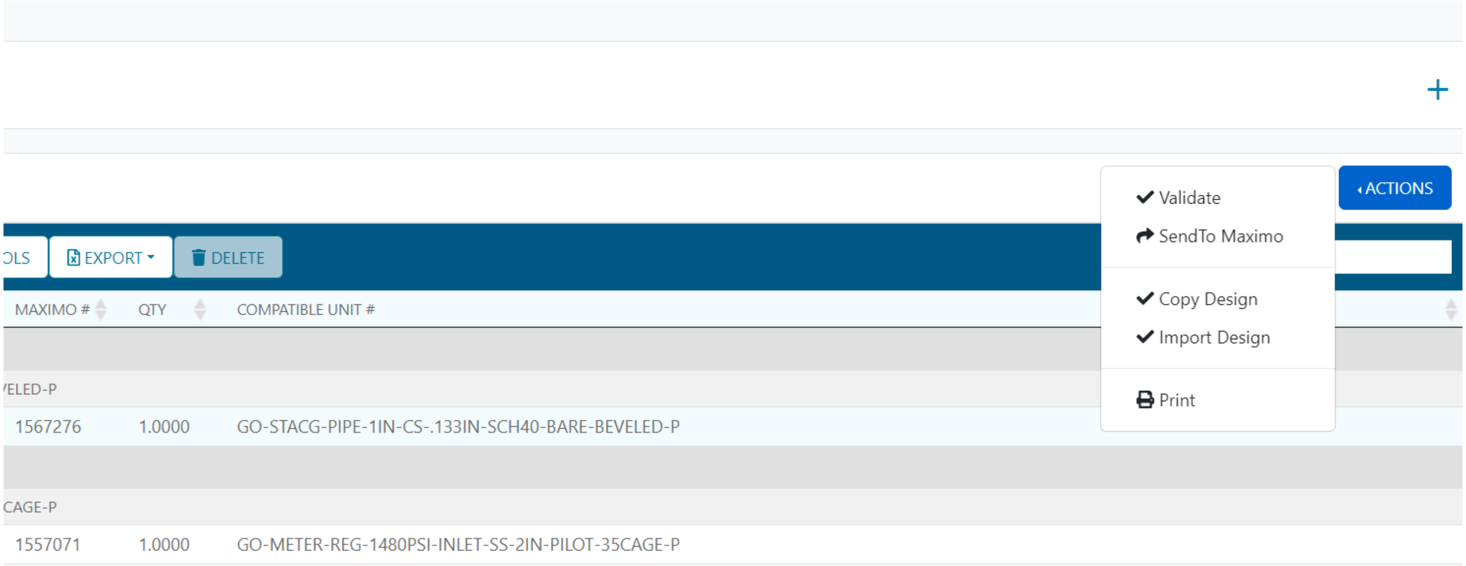
CADT - 1581

- BOM/Designs : Maximo Validation Update

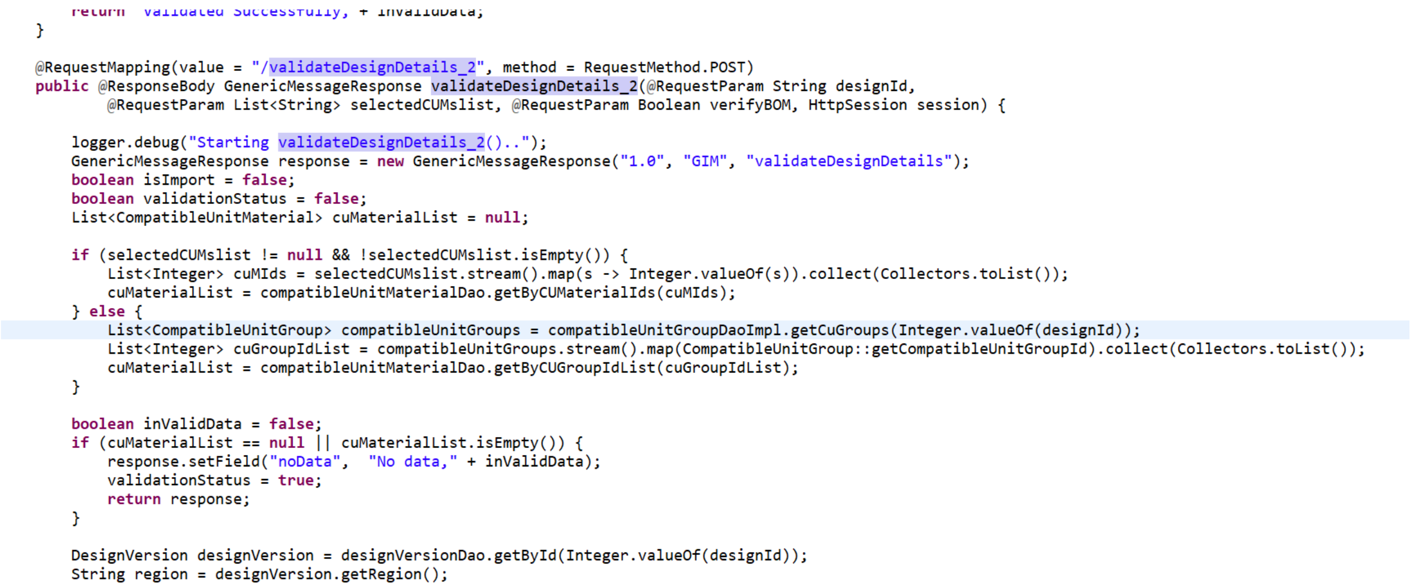
- Description: Maximo Validation should give detailed error message.
- Example :
  - Currently if no work order, just says “Validation complete with errors” but doesn’t list error anywhere
    - should list "No Work Order" in the "Validation Complete" pop up window

Analysis: When the click on any design and if in the design there is no workorder is selected if he tries to send to maximo the design then the popup model will come and inside that we must display that Validation complete with errors & No Work Order is selected.



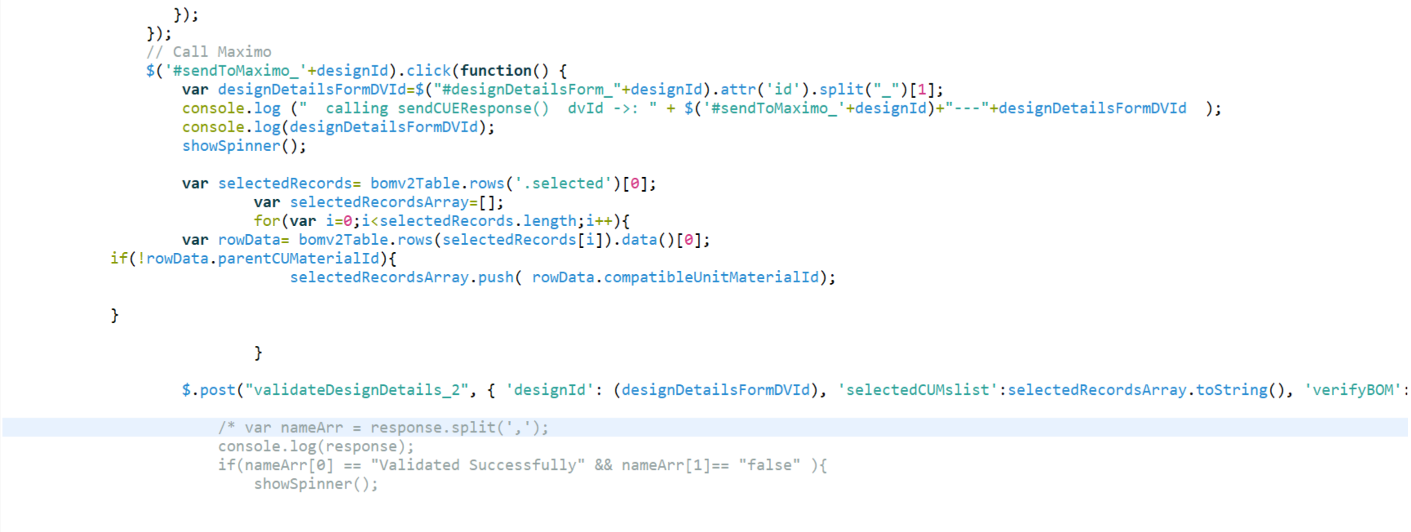
MAXIMO #	QTY	COMPATIBLE UNIT #
'ELED-P		
1567276	1.0000	GO-STACG-PIPE-1IN-CS-.133IN-SCH40-BARE-BEVELED-P
CAGE-P		
1557071	1.0000	GO-METER-REG-1480PSI-INLET-SS-2IN-PILOT-35CAGE-P

Conclusion: When the user click on send to maximo button it will trigger the Validation Controller (validateDesignDetails_2) and validateMaterials_ click function in Bomv2.html file



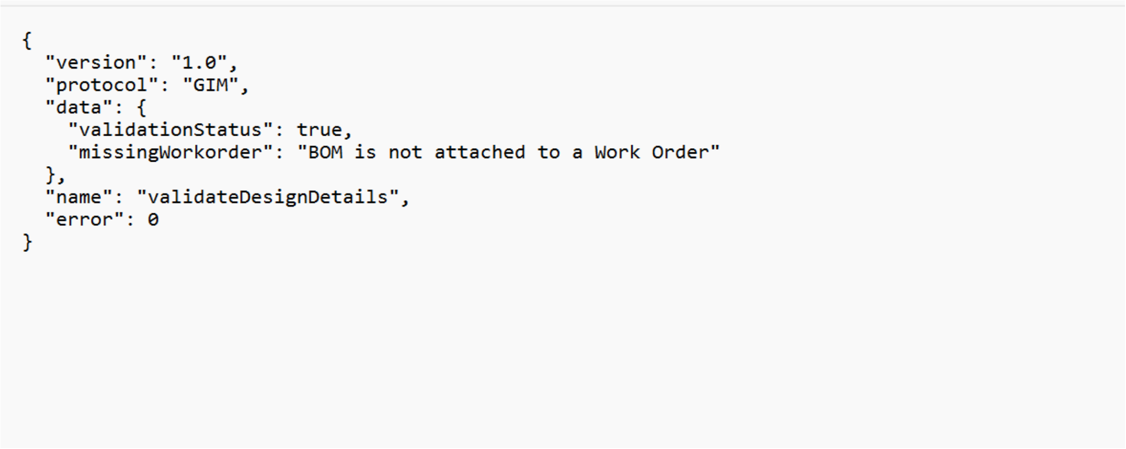
```
        return "Validated Successfully," + inValidData;
    }

    @RequestMapping(value = "/validateDesignDetails_2", method = RequestMethod.POST)
    public @ResponseBody GenericMessageResponse validateDesignDetails_2(@RequestParam String designId,
            @RequestParam List<String> selectedCUMslist, @RequestParam Boolean verifyBOM, HttpSession session) {

        logger.debug("Starting validateDesignDetails_2()..");
        GenericMessageResponse response = new GenericMessageResponse("1.0", "GIM", "validateDesignDetails");
        boolean isImport = false;
        boolean validationStatus = false;
        List<CompatibleUnitMaterial> cuMaterialList = null;

        if (selectedCUMslist != null && !selectedCUMslist.isEmpty()) {
            List<Integer> cuMIds = selectedCUMslist.stream().map(s -> Integer.valueOf(s)).collect(Collectors.toList());
            cuMaterialList = compatibleUnitMaterialDao.getByCUMaterialIds(cuMIds);
        } else {
            List<CompatibleUnitGroup> compatibleUnitGroups = compatibleUnitGroupDaoImpl.getCuGroups(Integer.valueOf(designId));
            List<Integer> cuGroupIdList = compatibleUnitGroups.stream().map(CompatibleUnitGroup::getCompatibleUnitGroupId).collect(Collectors.toList());
            cuMaterialList = compatibleUnitMaterialDao.getByCUGroupIdList(cuGroupIdList);
        }

        boolean inValidData = false;
        if (cuMaterialList == null || cuMaterialList.isEmpty()) {
            response.setField("noData",  "No data," + inValidData);
            validationStatus = true;
            return response;
        }

        DesignVersion designVersion = designVersionDao.getById(Integer.valueOf(designId));
        String region = designVersion.getRegion();
```

sendToMaximo_ click function



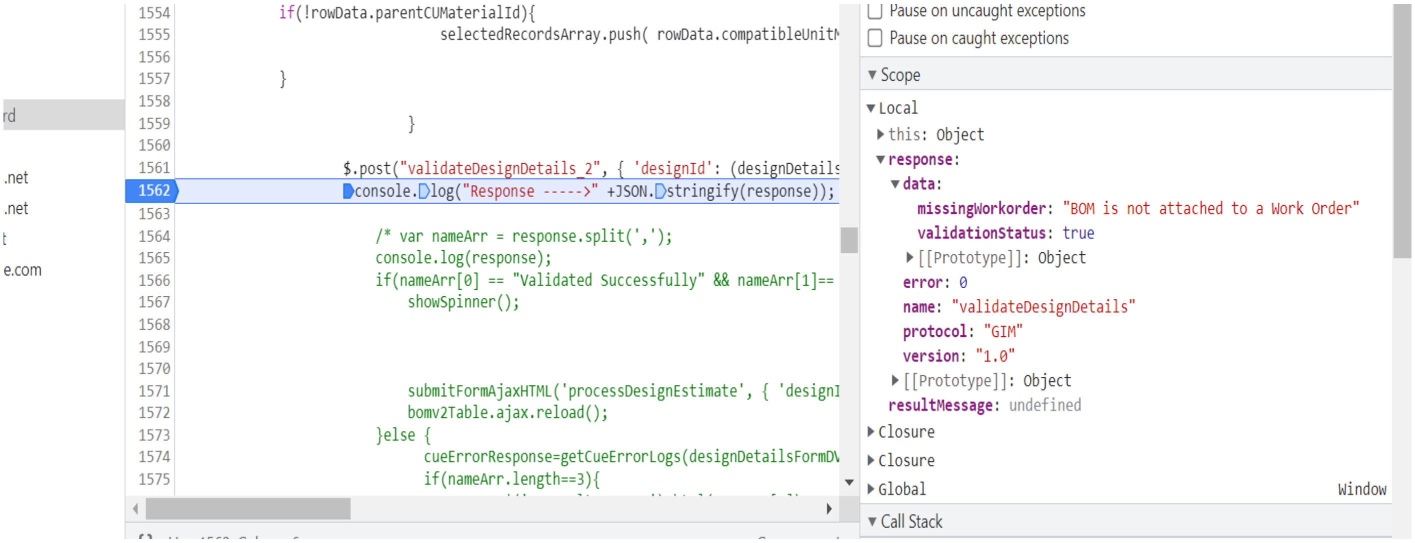
```
       });
    });
    // Call Maximo
    $('#sendToMaximo_'+designId).click(function() {
        var designDetailsFormDVId=$("#designDetailsForm_"+designId).attr('id').split("_")[1];
        console.log ("  calling sendCUEResponse()  dvId ->: " + $('#sendToMaximo_'+designId)+"---"+designDetailsFormDVId  );
        console.log(designDetailsFormDVId);
        showSpinner();

        var selectedRecords= bomv2Table.rows('.selected')[0];
                var selectedRecordsArray=[];
                for(var i=0;i<selectedRecords.length;i++){
        var rowData= bomv2Table.rows(selectedRecords[i]).data()[0];
if(!rowData.parentCUMaterialId){
                    selectedRecordsArray.push( rowData.compatibleUnitMaterialId);

}

                }

        $.post("validateDesignDetails_2", { 'designId': (designDetailsFormDVId), 'selectedCUMslist':selectedRecordsArray.toString(), 'verifyBOM':

            /* var nameArr = response.split(',');
            console.log(response);
            if(nameArr[0] == "Validated Successfully" && nameArr[1]== "false" ){
                showSpinner();
```

Now when we click on send To Maximo button we get an response data in the JSON format where you can find that missingworkorder == ‘BOM is not attached to a workorder’



```
{
  "version": "1.0",
  "protocol": "GIM",
  "data": {
    "validationStatus": true,
    "missingWorkorder": "BOM is not attached to a Work Order"
  },
  "name": "validateDesignDetails",
  "error": 0
}
```

Now you can also get this by putting the debug point where this Maximo validation is called inside $.post() call put the debug point and you easily identify from there that data is showing missingworkorder which is equal to BOM is not attached to the work order.



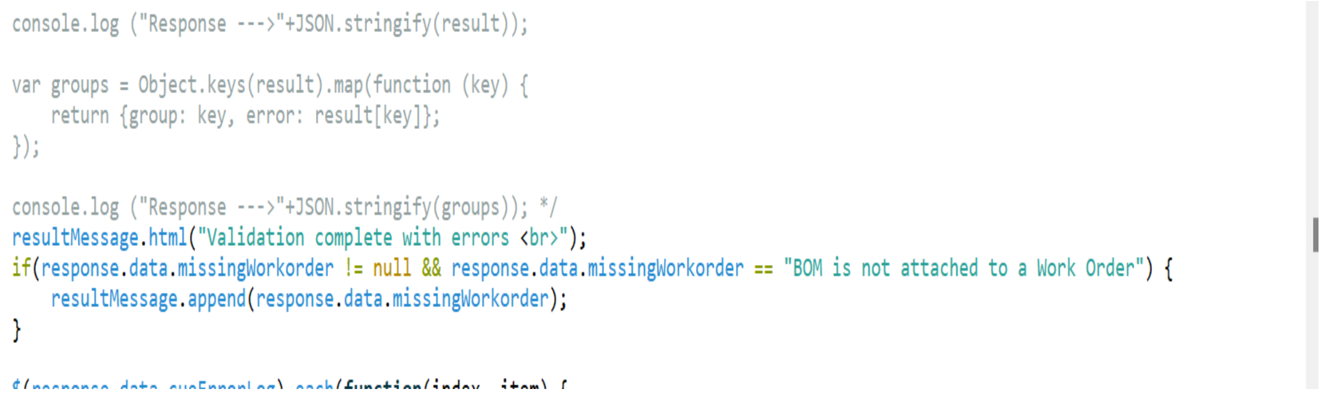
```
            if(!rowData.parentCUMaterialId){
                        selectedRecordsArray.push( rowData.compatibleUnitM

            }

                    }

                $.post("validateDesignDetails_2", { 'designId': (designDetails
                console.log("Response ----->" +JSON.stringify(response));

                    /* var nameArr = response.split(',');
                    console.log(response);
                    if(nameArr[0] == "Validated Successfully" && nameArr[1]==
                        showSpinner();



                        submitFormAjaxHTML('processDesignEstimate', { 'designI
                        bomv2Table.ajax.reload();
                    }else {
                        cueErrorResponse=getCueErrorLogs(designDetailsFormDV
                        if(nameArr.length==3){
```

□ Pause on uncaught exceptions

□ Pause on caught exceptions

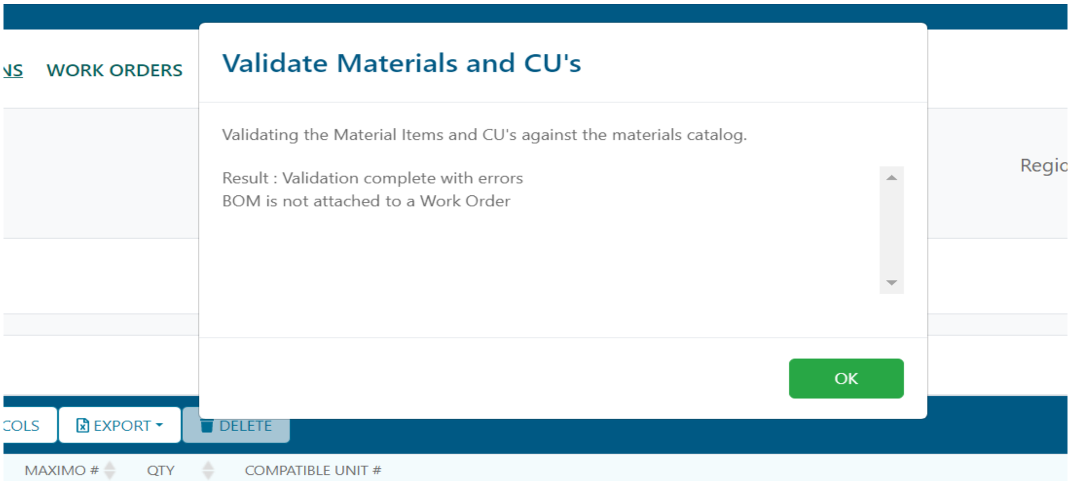
```
▼ Scope
▼ Local
  ▶ this: Object
  ▼ response:
    ▼ data:
        missingWorkorder: "BOM is not attached to a Work Order"
        validationStatus: true
      ▶ [[Prototype]]: Object
      error: 0
      name: "validateDesignDetails"
      protocol: "GIM"
      version: "1.0"
    ▶ [[Prototype]]: Object
  resultMessage: undefined
▶ Closure
▶ Closure
▶ Global                                          Window
▼ Call Stack
```

Solution: Now we can append the Missingworkorder value in the popup model whenever user click on send to maximo button if the workorder is null then it will display that ‘BOM is not attached to the Work Order’



```
console.log ("Response --->"+JSON.stringify(result));

var groups = Object.keys(result).map(function (key) {
    return {group: key, error: result[key]};
});

console.log ("Response --->"+JSON.stringify(groups)); */
resultMessage.html("Validation complete with errors <br>");
if(response.data.missingWorkorder != null && response.data.missingWorkorder == "BOM is not attached to a Work Order") {
    resultMessage.append(response.data.missingWorkorder);
}

$(response.data.sueErrorLog).each(function(index, item) {
```

Result: It will display the result like this as mentioned below.



Validate Materials and CU's

Validating the Material Items and CU's against the materials catalog.

Result : Validation complete with errors
BOM is not attached to a Work Order