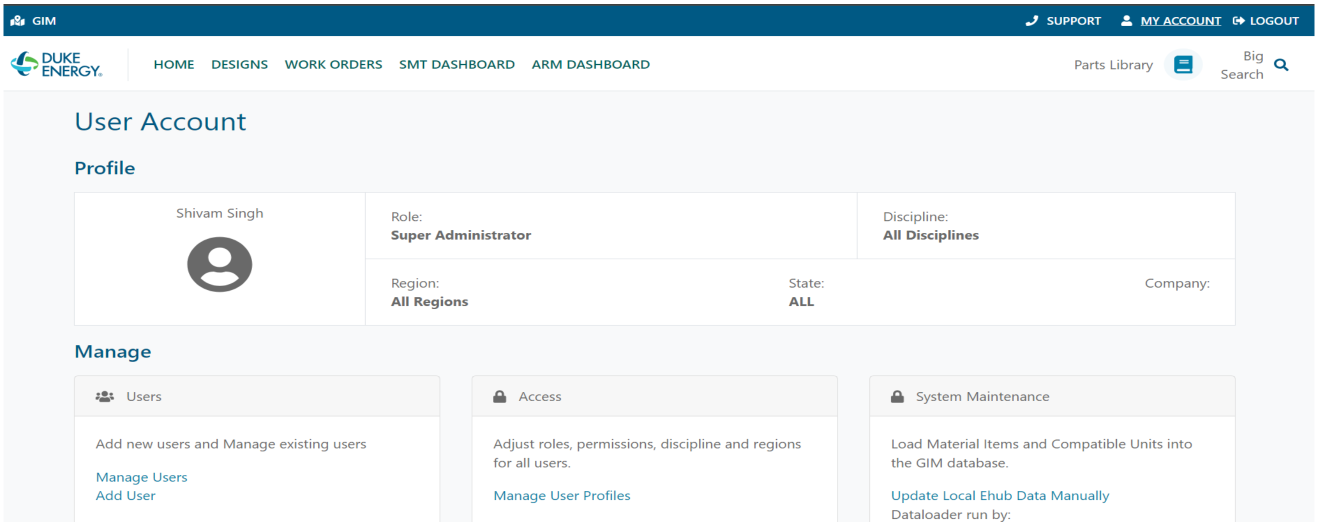
CADT – 1480
BOM/Designs : Group Changes

- Super Admin users should be able to add/remove groups & update the names for existing Groups to accommodate group change needs
    - Current group list includes : Major Components, Fittings, Flanges, Valves, Miscellaneous, Materials for Test Connections, Materials for Strainer Blowdowns, Materials for Regulator Sense Lines, Pipe, Long Lead
    - List should be editable
        - Potentially add a new section to "My Account" page. Section would contain a list of the group names. List would allow a Super Admin user to add/remove/edit.
        - Need to make accommodations for what happens to group names that have been remove
- Create a "Default" list of groups that user could select to add automatically to BOM
    - list of default groups to be provided - Brittany will get from Janel

Analysis: Here what I have analyzed that we have to create an card inside myAccountPage and that card should be visible to SuperAdmin and GimAdministrator user so that they can edit and add new groups and also ability to delete any groups as well so here the frontend change and as well as Backend changes are included .



User Account

Profile

Shivam Singh

Role:
Super Administrator

Discipline:
All Disciplines

Region:
All Regions

State:
ALL

Company:

Manage

Users

Add new users and Manage existing users

Manage Users
Add User

Access

Adjust roles, permissions, discipline and regions for all users.

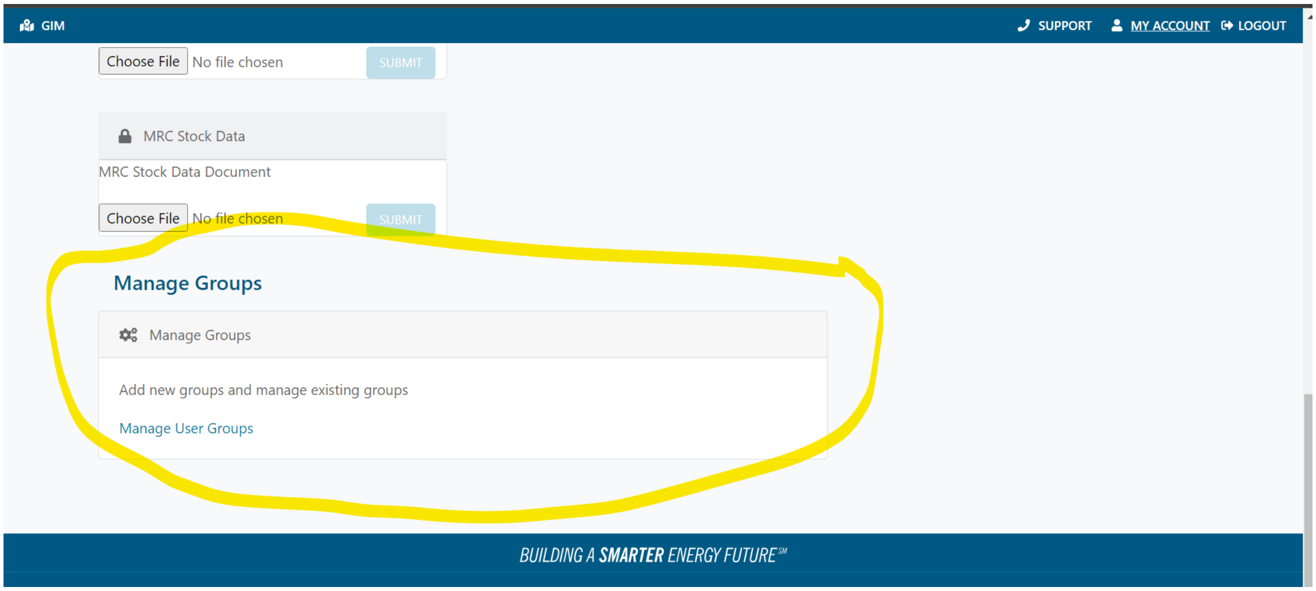
Manage User Profiles

System Maintenance

Load Material Items and Compatible Units into the GIM database.

Update Local Ehub Data Manually
Dataloader run by:

FrontEndchanges: So here I have to create this type of Card inside the Account page and here when the user click on Manage User Groups then it will load the datatable with having all the groups content.



MRC Stock Data Document

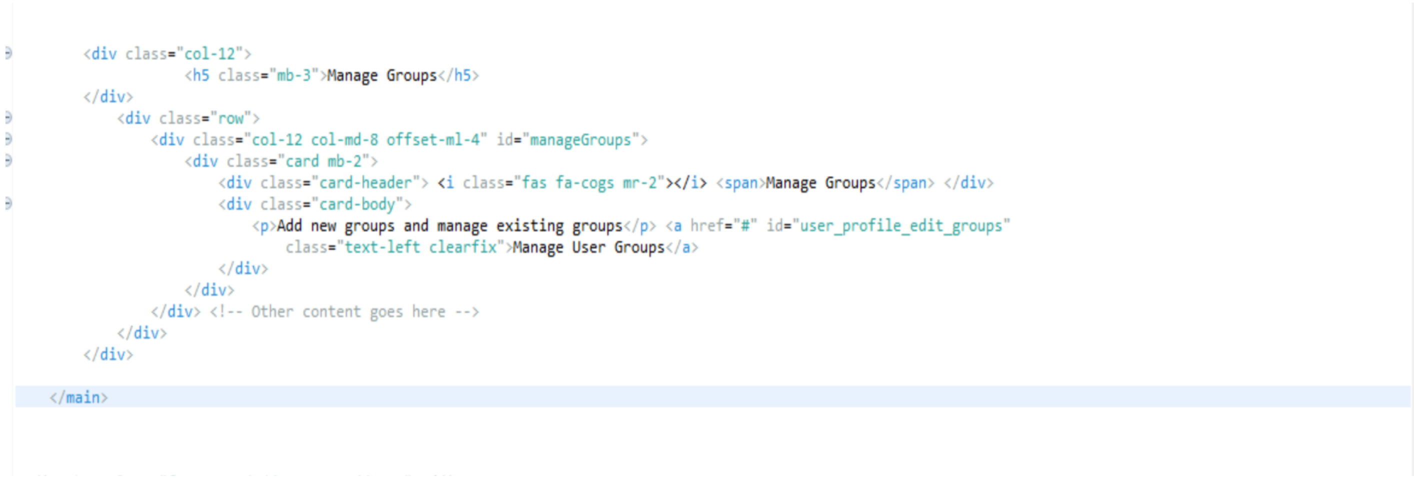
Manage Groups

Manage Groups

Add new groups and manage existing groups

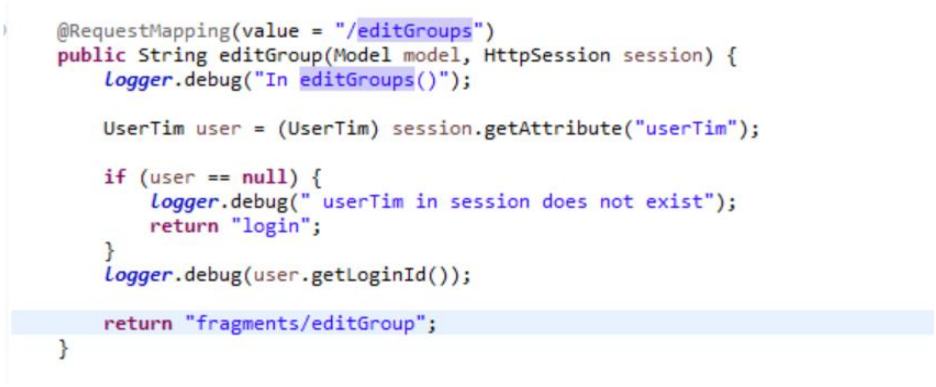
Manage User Groups

Step 1: In this Profile.html file I have created this card with appropriate headings and also gave the id="user_profile_edit_groups"

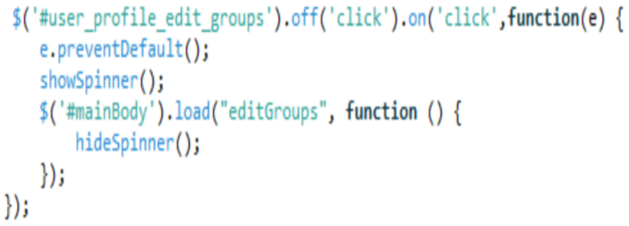


```
        <div class="col-12">
                <h5 class="mb-3">Manage Groups</h5>
        </div>
            <div class="row">
                <div class="col-12 col-md-8 offset-ml-4" id="manageGroups">
                    <div class="card mb-2">
                        <div class="card-header"> <i class="fas fa-cogs mr-2"></i> <span>Manage Groups</span> </div>
                        <div class="card-body">
                            <p>Add new groups and manage existing groups</p> <a href="#" id="user_profile_edit_groups"
                                class="text-left clearfix">Manage User Groups</a>
                        </div>
                    </div>
                </div> <!-- Other content goes here -->
            </div>
        </div>

    </main>
```

Step 2: So here what I have did when user click on that link it will trigger here and it will show spinner and it will trigger the controller with having endpoints /editGroups as shown below:

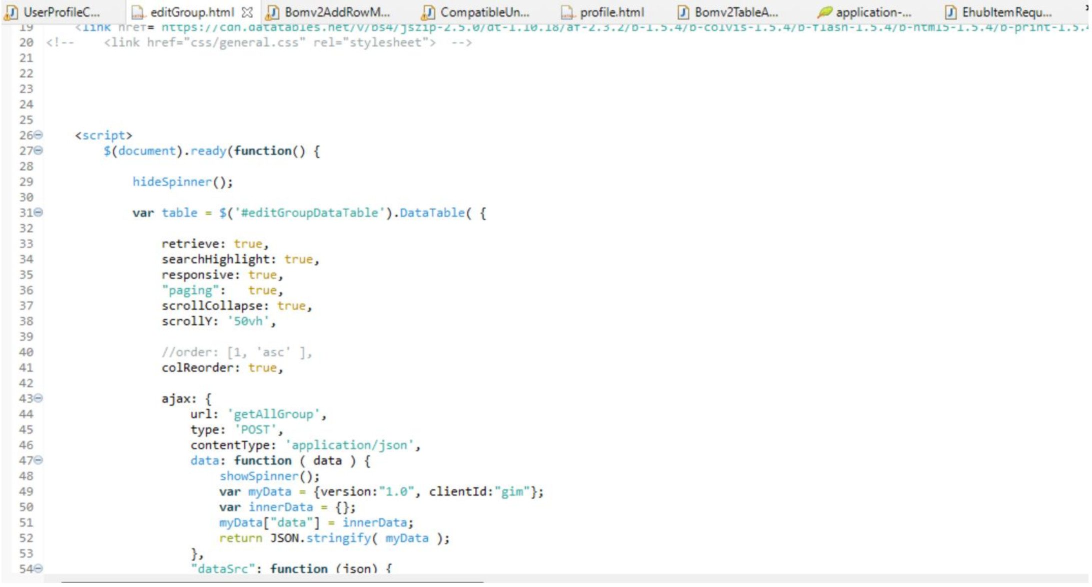


```
@RequestMapping(value = "/editGroups")
public String editGroup(Model model, HttpSession session) {
    logger.debug("In editGroups()");

    UserTim user = (UserTim) session.getAttribute("userTim");

    if (user == null) {
        logger.debug(" userTim in session does not exist");
        return "login";
    }
    logger.debug(user.getLoginId());

    return "fragments/editGroup";
}
```

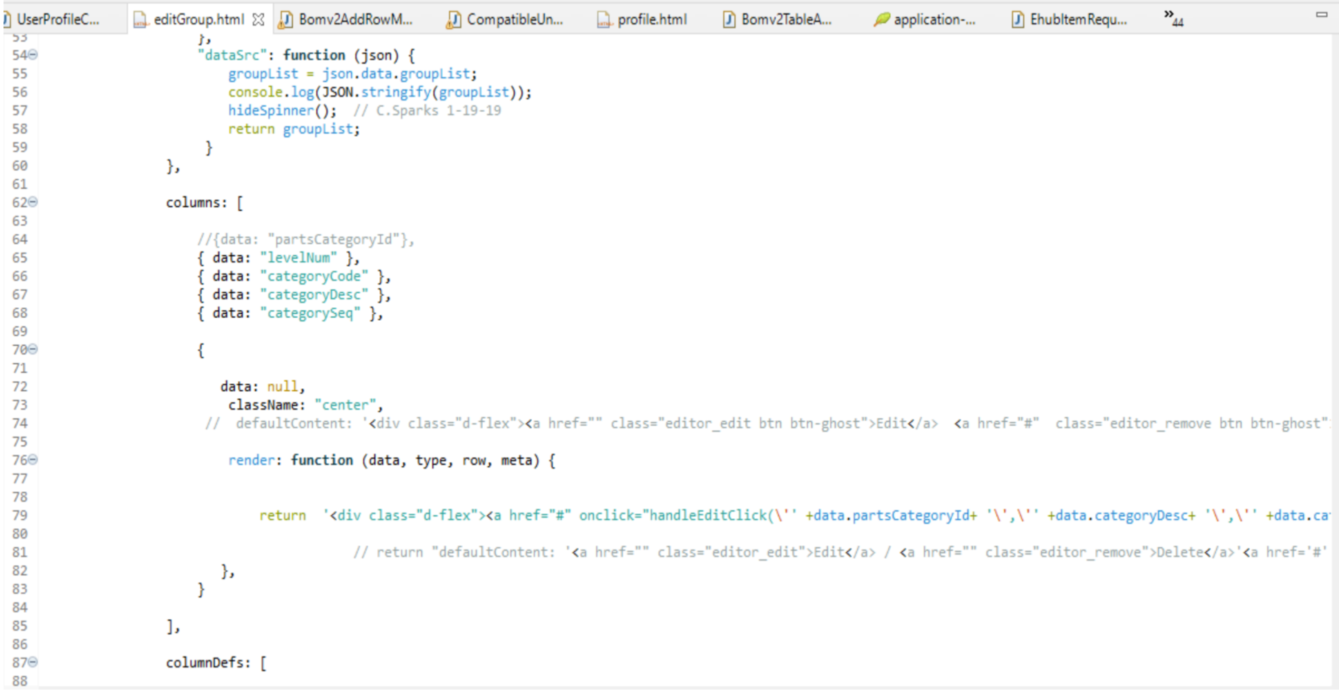


```
$('#user_profile_edit_groups').off('click').on('click',function(e) {
    e.preventDefault();
    showSpinner();
    $('#mainBody').load("editGroups", function () {
        hideSpinner();
    });
});
```

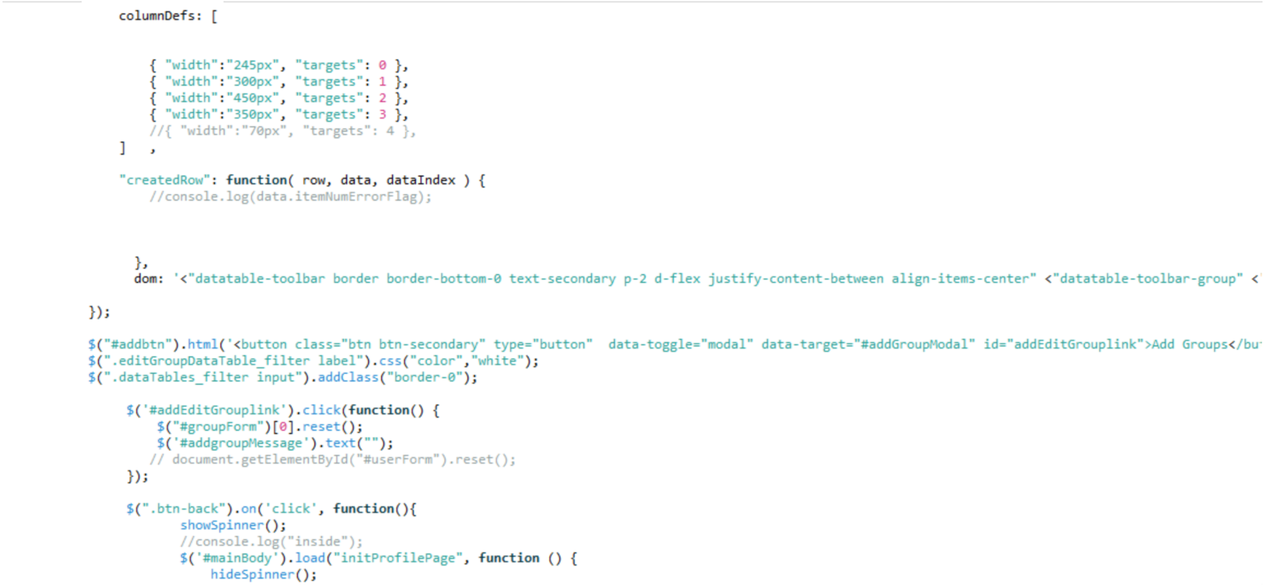
Step 3:Here this is the editGroup.html page I have created in order to create the datatable.



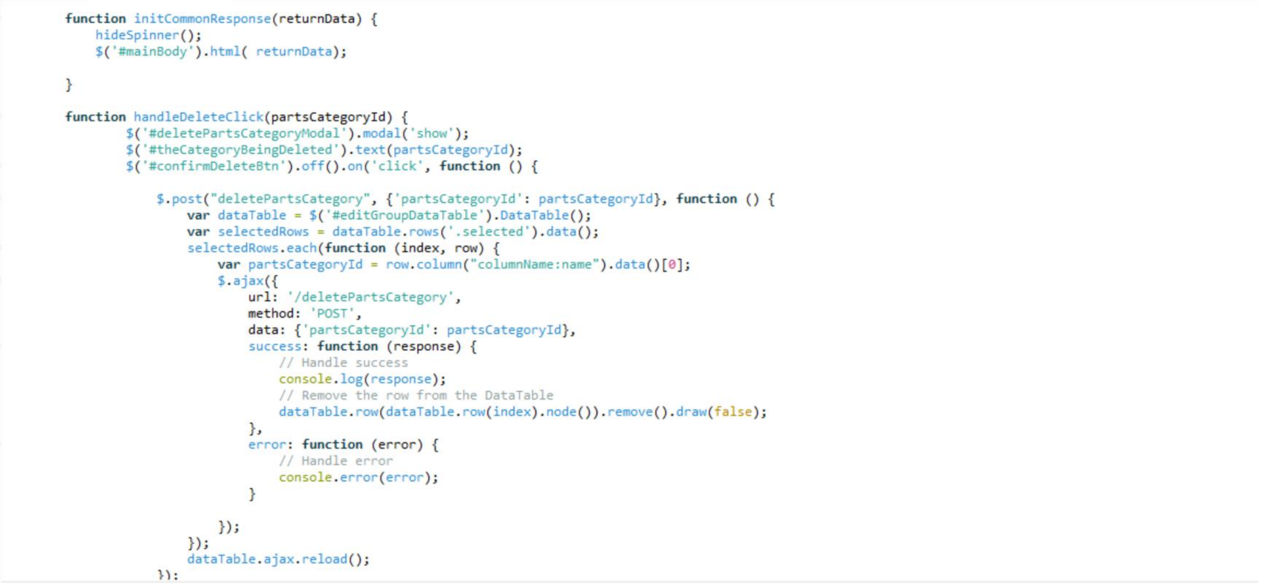
```
    <link href= https://cdn.datatables.net/v/bs4/jszip-2.5.0/dt-1.10.18/af-2.3.2/b-1.5.4/b-colvis-1.5.4/b-flash-1.5.4/b-html5-1.5.4/b-print-1.5.
<!--    <link href="css/general.css" rel="stylesheet">  -->





    <script>
        $(document).ready(function() {

            hideSpinner();

            var table = $('#editGroupDataTable').DataTable( {

                retrieve: true,
                searchHighlight: true,
                responsive: true,
                "paging":   true,
                scrollCollapse: true,
                scrollY: '50vh',

                //order: [1, 'asc' ],
                colReorder: true,

                ajax: {
                    url: 'getAllGroup',
                    type: 'POST',
                    contentType: 'application/json',
                    data: function ( data ) {
                        showSpinner();
                        var myData = {version:"1.0", clientId:"gim"};
                        var innerData = {};
                        myData["data"] = innerData;
                        return JSON.stringify( myData );
                    },
                    "dataSrc": function (json) {
```



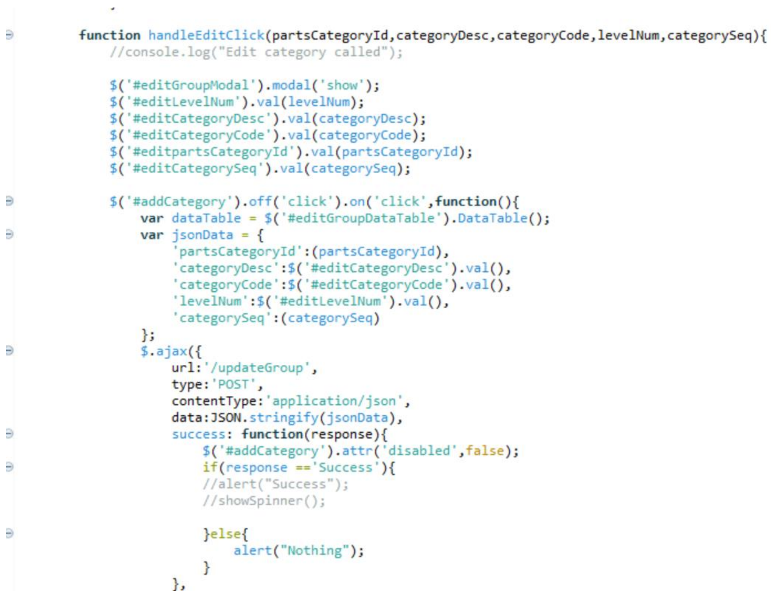
```
            },
            "dataSrc": function (json) {
                groupList = json.data.groupList;
                console.log(JSON.stringify(groupList));
                hideSpinner();  // C.Sparks 1-19-19
                return groupList;
             }
        },

        columns: [

            //{data: "partsCategoryId"},
            { data: "levelNum" },
            { data: "categoryCode" },
            { data: "categoryDesc" },
            { data: "categorySeq" },

            {

               data: null,
                className: "center",
             //  defaultContent: '<div class="d-flex"><a href="" class="editor_edit btn btn-ghost">Edit</a>  <a href="#"  class="editor_remove btn btn-ghost"

                render: function (data, type, row, meta) {


                    return  '<div class="d-flex"><a href="#" onclick="handleEditClick(\'' +data.partsCategoryId+ '\',\'' +data.categoryDesc+ '\',\'' +data.ca

                                // return "defaultContent: '<a href="" class="editor_edit">Edit</a> / <a href="" class="editor_remove">Delete</a>'<a href='#'
               },
            }

        ],

        columnDefs: [
```



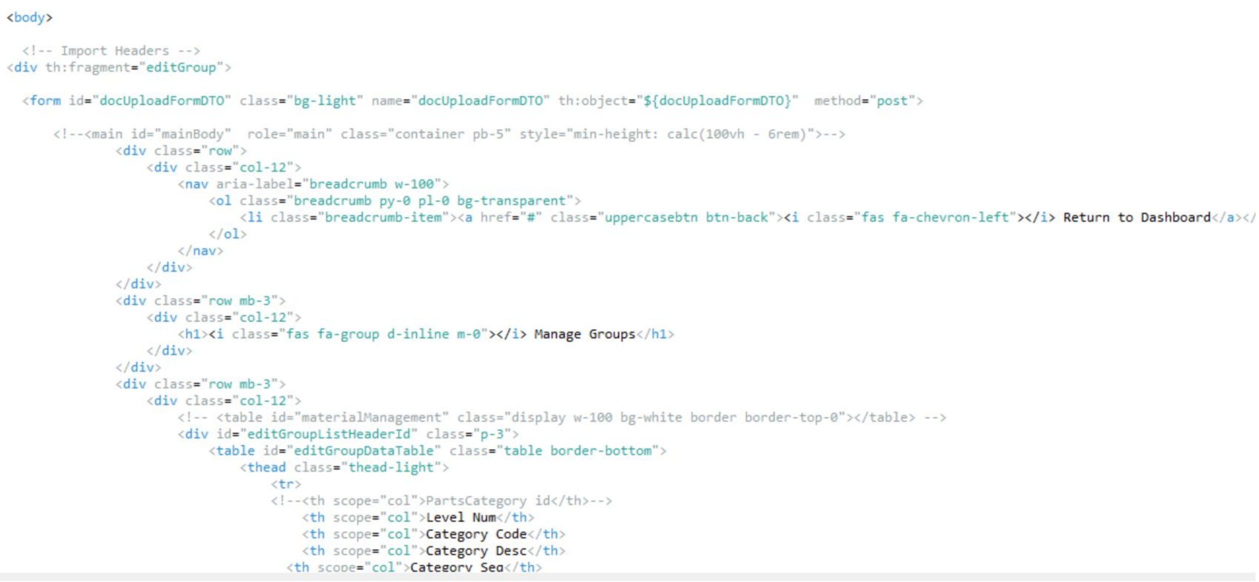
```
        columnDefs: [


            { "width":"245px", "targets": 0 },
            { "width":"300px", "targets": 1 },
            { "width":"450px", "targets": 2 },
            { "width":"350px", "targets": 3 },
            //{ "width":"70px", "targets": 4 },
        ]   ,

        "createdRow": function( row, data, dataIndex ) {
            //console.log(data.itemNumErrorFlag);



          },
          dom: '<"datatable-toolbar border border-bottom-0 text-secondary p-2 d-flex justify-content-between align-items-center" <"datatable-toolbar-group" <

    });

    $("#addbtn").html('<button class="btn btn-secondary" type="button"  data-toggle="modal" data-target="#addGroupModal" id="addEditGrouplink">Add Groups</bu
    $(".editGroupDataTable_filter label").css("color","white");
    $(".dataTables_filter input").addClass("border-0");

         $('#addEditGrouplink').click(function() {
             $("#groupForm")[0].reset();
             $('#addgroupMessage').text("");
            // document.getElementById("#userForm").reset();
         });

         $(".btn-back").on('click', function(){
                showSpinner();
                //console.log("inside");
                $('#mainBody').load("initProfilePage", function () {
                    hideSpinner();
```



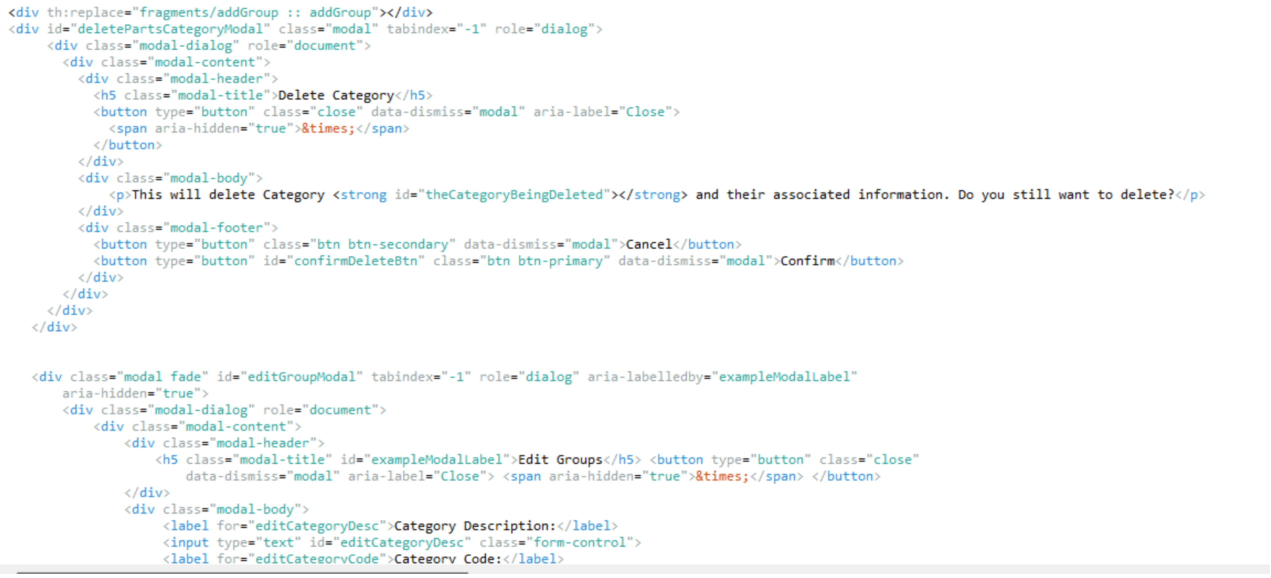
```
function initCommonResponse(returnData) {
    hideSpinner();
    $('#mainBody').html( returnData);

}

function handleDeleteClick(partsCategoryId) {
        $('#deletePartsCategoryModal').modal('show');
        $('#theCategoryBeingDeleted').text(partsCategoryId);
        $('#confirmDeleteBtn').off().on('click', function () {

            $.post("deletePartsCategory", {'partsCategoryId': partsCategoryId}, function () {
                var dataTable = $('#editGroupDataTable').DataTable();
                var selectedRows = dataTable.rows('.selected').data();
                selectedRows.each(function (index, row) {
                    var partsCategoryId = row.column("columnName:name").data()[0];
                    $.ajax({
                        url: '/deletePartsCategory',
                        method: 'POST',
                        data: {'partsCategoryId': partsCategoryId},
                        success: function (response) {
                            // Handle success
                            console.log(response);
                            // Remove the row from the DataTable
                            dataTable.row(dataTable.row(index).node()).remove().draw(false);
                        },
                        error: function (error) {
                            // Handle error
                            console.error(error);
                        }

                    });
                });
                dataTable.ajax.reload();
            });
```



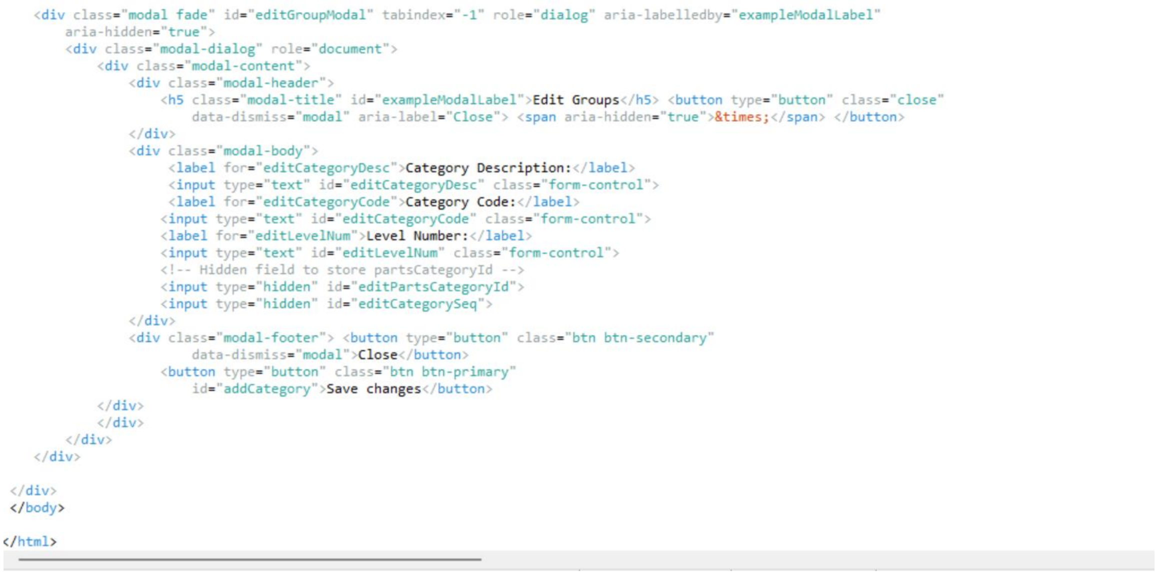
```
function handleEditClick(partsCategoryId,categoryDesc,categoryCode,levelNum,categorySeq){
    //console.log("Edit category called");

    $('#editGroupModal').modal('show');
    $('#editLevelNum').val(levelNum);
    $('#editCategoryDesc').val(categoryDesc);
    $('#editCategoryCode').val(categoryCode);
    $('#editpartsCategoryId').val(partsCategoryId);
    $('#editCategorySeq').val(categorySeq);

    $('#addCategory').off('click').on('click',function(){
        var dataTable = $('#editGroupDataTable').DataTable();
        var jsonData = {
            'partsCategoryId':(partsCategoryId),
            'categoryDesc':$('#editCategoryDesc').val(),
            'categoryCode':$('#editCategoryCode').val(),
            'levelNum':$('#editLevelNum').val(),
            'categorySeq':(categorySeq)
        };
        $.ajax({
            url:'/updateGroup',
            type:'POST',
            contentType:'application/json',
            data:JSON.stringify(jsonData),
            success: function(response){
                $('#addCategory').attr('disabled',false);
                if(response =='Success'){
                //alert("Success");
                //showSpinner();

                }else{
                    alert("Nothing");
                }
            },
```



```
<body>

  <!-- Import Headers -->
<div th:fragment="editGroup">

  <form id="docUploadFormDTO" class="bg-light" name="docUploadFormDTO" th:object="${docUploadFormDTO}"  method="post">

      <!--<main id="mainBody"  role="main" class="container pb-5" style="min-height: calc(100vh - 6rem)">-->
              <div class="row">
                  <div class="col-12">
                      <nav aria-label="breadcrumb w-100">
                          <ol class="breadcrumb py-0 pl-0 bg-transparent">
                              <li class="breadcrumb-item"><a href="#" class="uppercasebtn btn-back"><i class="fas fa-chevron-left"></i> Return to Dashboard</a></
                          </ol>
                      </nav>
                  </div>
              </div>
              <div class="row mb-3">
                  <div class="col-12">
                      <h1><i class="fas fa-group d-inline m-0"></i> Manage Groups</h1>
                  </div>
              </div>
              <div class="row mb-3">
                  <div class="col-12">
                      <!-- <table id="materialManagement" class="display w-100 bg-white border border-top-0"></table> -->
                      <div id="editGroupListHeaderId" class="p-3">
                          <table id="editGroupDataTable" class="table border-bottom">
                              <thead class="thead-light">
                                  <tr>
                                  <!--<th scope="col">PartsCategory id</th>-->
                                      <th scope="col">Level Num</th>
                                      <th scope="col">Category Code</th>
                                      <th scope="col">Category Desc</th>
                                    <th scope="col">Category Seq</th>
```

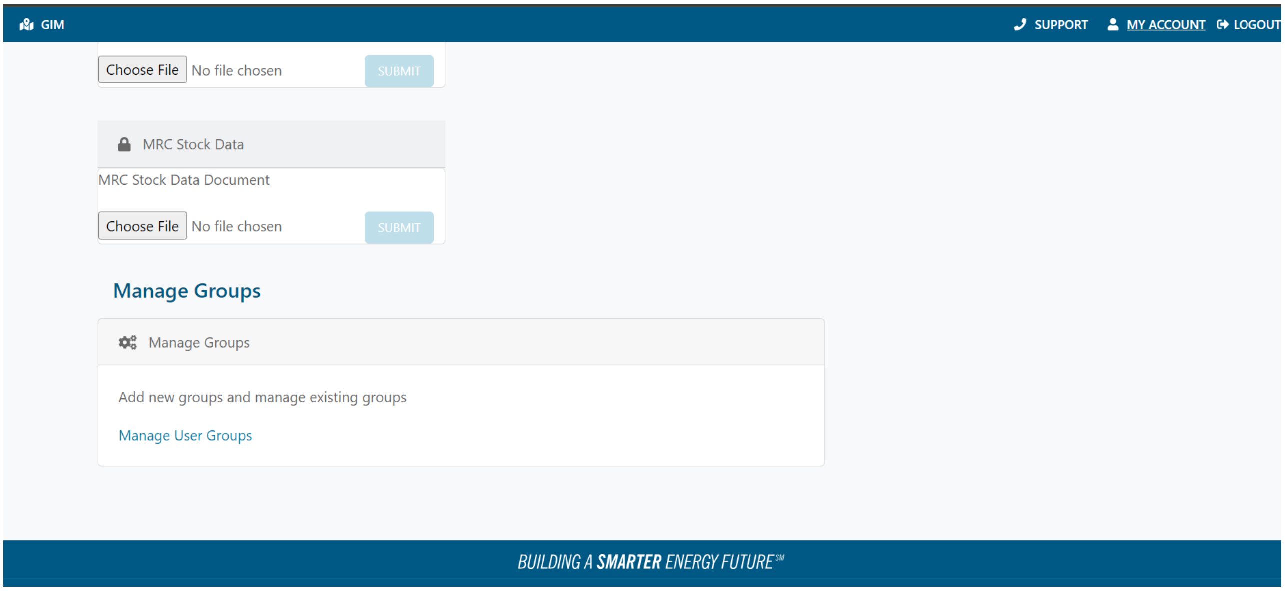


```
<div th:replace="fragments/addGroup :: addGroup"></div>
<div id="deletePartsCategoryModal" class="modal" tabindex="-1" role="dialog">
     <div class="modal-dialog" role="document">
       <div class="modal-content">
         <div class="modal-header">
           <h5 class="modal-title">Delete Category</h5>
           <button type="button" class="close" data-dismiss="modal" aria-label="Close">
             <span aria-hidden="true">&times;</span>
           </button>
         </div>
         <div class="modal-body">
             <p>This will delete Category <strong id="theCategoryBeingDeleted"></strong> and their associated information. Do you still want to delete?</p>
         </div>
         <div class="modal-footer">
           <button type="button" class="btn btn-secondary" data-dismiss="modal">Cancel</button>
           <button type="button" id="confirmDeleteBtn" class="btn btn-primary" data-dismiss="modal">Confirm</button>
         </div>
       </div>
     </div>
   </div>


   <div class="modal fade" id="editGroupModal" tabindex="-1" role="dialog" aria-labelledby="exampleModalLabel"
       aria-hidden="true">
       <div class="modal-dialog" role="document">
           <div class="modal-content">
               <div class="modal-header">
                   <h5 class="modal-title" id="exampleModalLabel">Edit Groups</h5> <button type="button" class="close"
                       data-dismiss="modal" aria-label="Close"> <span aria-hidden="true">&times;</span> </button>
               </div>
               <div class="modal-body">
                    <label for="editCategoryDesc">Category Description:</label>
                    <input type="text" id="editCategoryDesc" class="form-control">
                    <label for="editCategoryCode">Category Code:</label>
```



```
    <div class="modal fade" id="editGroupModal" tabindex="-1" role="dialog" aria-labelledby="exampleModalLabel"
        aria-hidden="true">
        <div class="modal-dialog" role="document">
            <div class="modal-content">
                <div class="modal-header">
                    <h5 class="modal-title" id="exampleModalLabel">Edit Groups</h5> <button type="button" class="close"
                        data-dismiss="modal" aria-label="Close"> <span aria-hidden="true">&times;</span> </button>
                </div>
                <div class="modal-body">
                     <label for="editCategoryDesc">Category Description:</label>
                     <input type="text" id="editCategoryDesc" class="form-control">
                     <label for="editCategoryCode">Category Code:</label>
                    <input type="text" id="editCategoryCode" class="form-control">
                    <label for="editLevelNum">Level Number:</label>
                    <input type="text" id="editLevelNum" class="form-control">
                    <!-- Hidden field to store partsCategoryId -->
                    <input type="hidden" id="editPartsCategoryId">
                    <input type="hidden" id="editCategorySeq">
                </div>
                <div class="modal-footer"> <button type="button" class="btn btn-secondary"
                        data-dismiss="modal">Close</button>
                    <button type="button" class="btn btn-primary"
                        id="addCategory">Save changes</button>
            </div>
            </div>
        </div>
    </div>

 </div>
 </body>

</html>
```

- So here this is the Card structure is ready with having Manage Groups



MRC Stock Data

MRC Stock Data Document

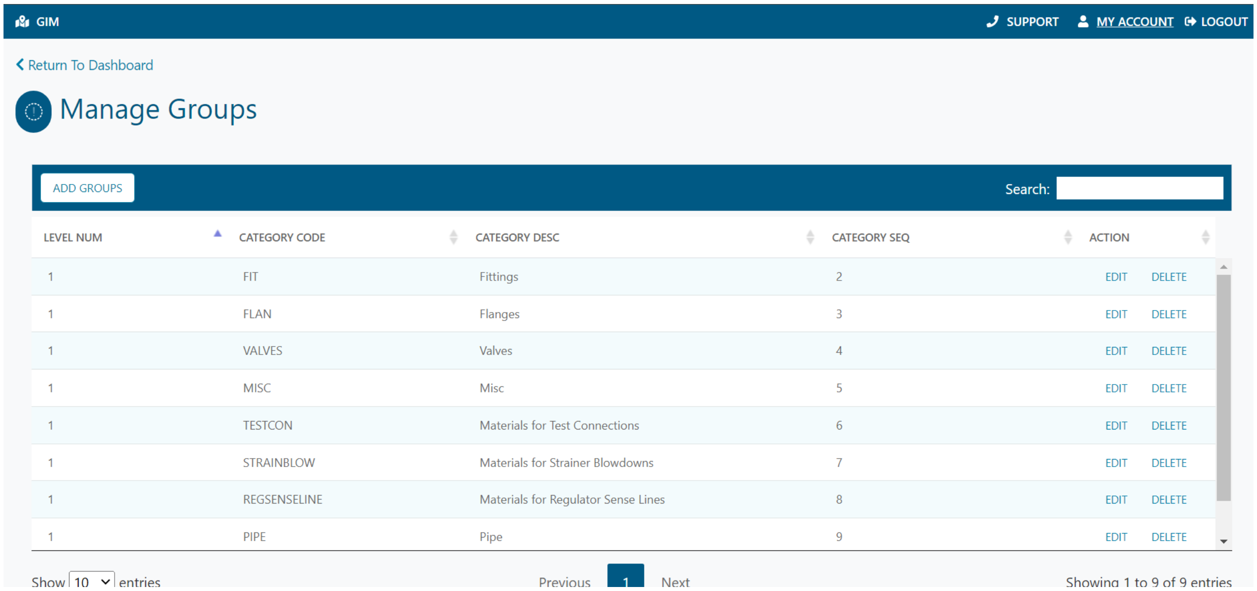
Manage Groups

Manage Groups

Add new groups and manage existing groups

Manage User Groups

- So here this is the datatable is successfully created with the Group data.



Manage Groups

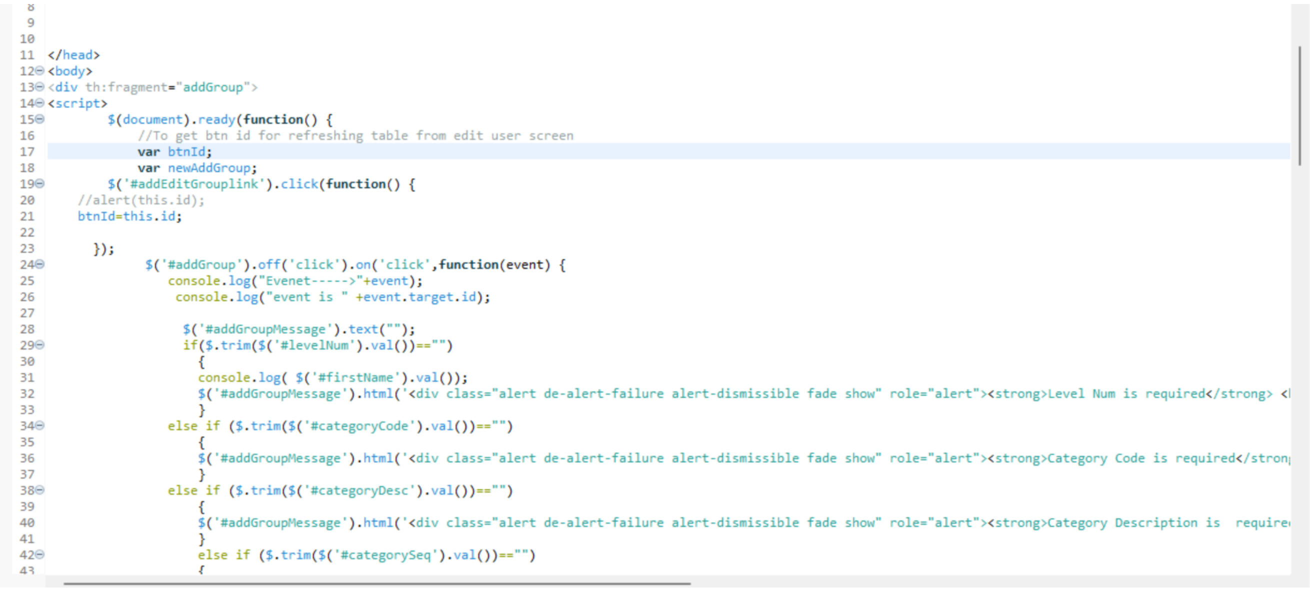
LEVEL NUM	CATEGORY CODE	CATEGORY DESC	CATEGORY SEQ	ACTION
1	FIT	Fittings	2	EDIT DELETE
1	FLAN	Flanges	3	EDIT DELETE
1	VALVES	Valves	4	EDIT DELETE
1	MISC	Misc	5	EDIT DELETE
1	TESTCON	Materials for Test Connections	6	EDIT DELETE
1	STRAINBLOW	Materials for Strainer Blowdowns	7	EDIT DELETE
1	REGSENSELINE	Materials for Regulator Sense Lines	8	EDIT DELETE
1	PIPE	Pipe	9	EDIT DELETE

Show 10 entries

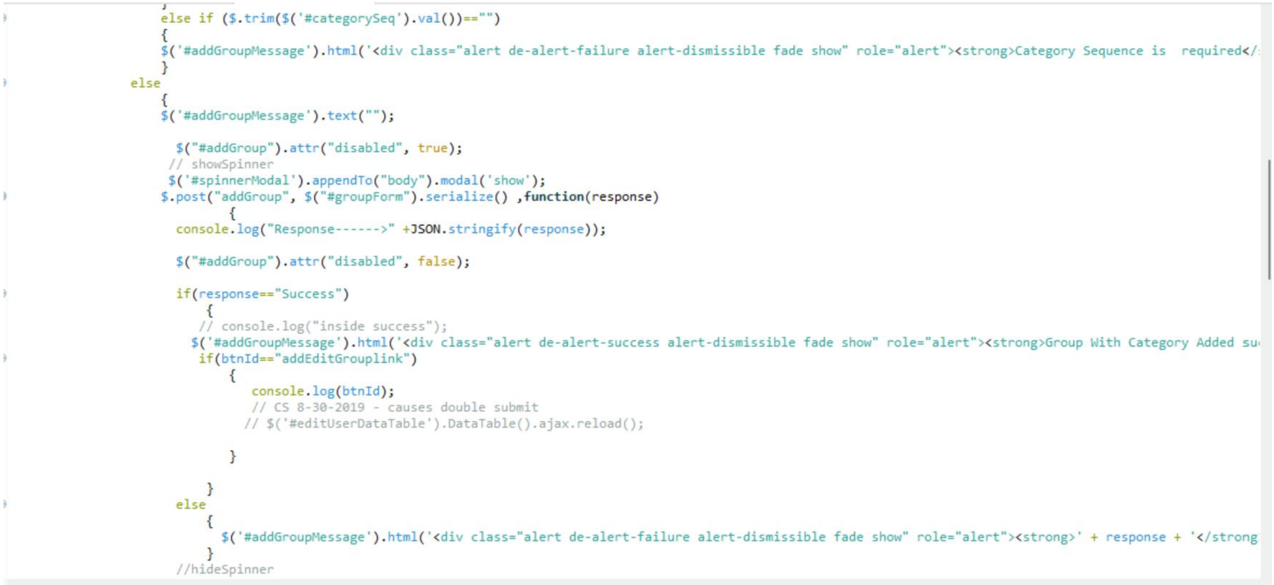
Previous 1 Next

Showing 1 to 9 of 9 entries

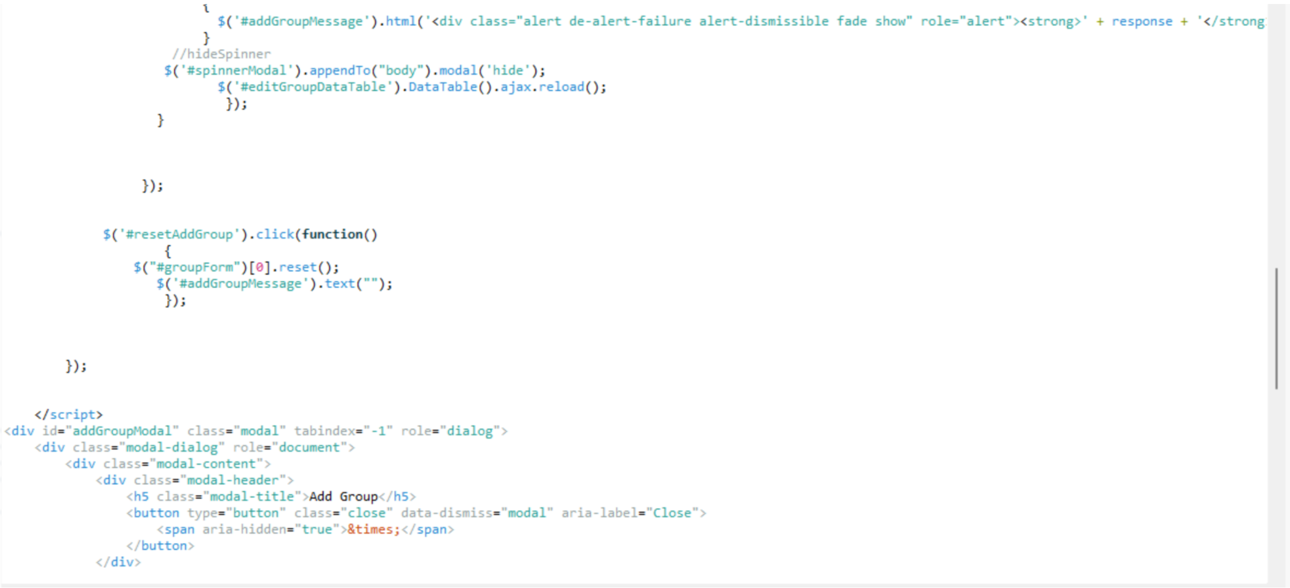
- So here I have created an addGroup.html file in order to create the modal popup for AddGroup with having different fields and having validation of the fields has well as shown in the screenshot below:



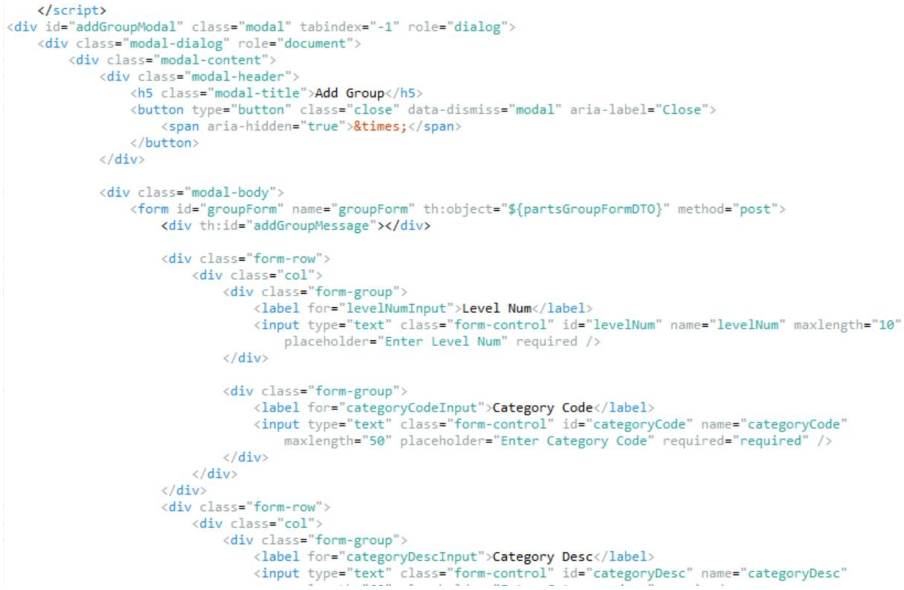
```



</head>
<body>
<div th:fragment="addGroup">
<script>
        $(document).ready(function() {
            //To get btn id for refreshing table from edit user screen
            var btnId;
            var newAddGroup;
        $('#addEditGrouplink').click(function() {
    //alert(this.id);
    btnId=this.id;

      });
             $('#addGroup').off('click').on('click',function(event) {
                console.log("Evenet----->"+event);
                 console.log("event is " +event.target.id);

                  $('#addGroupMessage').text("");
                  if($.trim($('#levelNum').val())=="")
                    {
                    console.log( $('#firstName').val());
                    $('#addGroupMessage').html('<div class="alert de-alert-failure alert-dismissible fade show" role="alert"><strong>Level Num is required</strong> <
                    }
                else if ($.trim($('#categoryCode').val())=="")
                    {
                    $('#addGroupMessage').html('<div class="alert de-alert-failure alert-dismissible fade show" role="alert"><strong>Category Code is required</stron
                    }
                else if ($.trim($('#categoryDesc').val())=="")
                    {
                    $('#addGroupMessage').html('<div class="alert de-alert-failure alert-dismissible fade show" role="alert"><strong>Category Description is  require
                    }
                    else if ($.trim($('#categorySeq').val())=="")
                    {
```



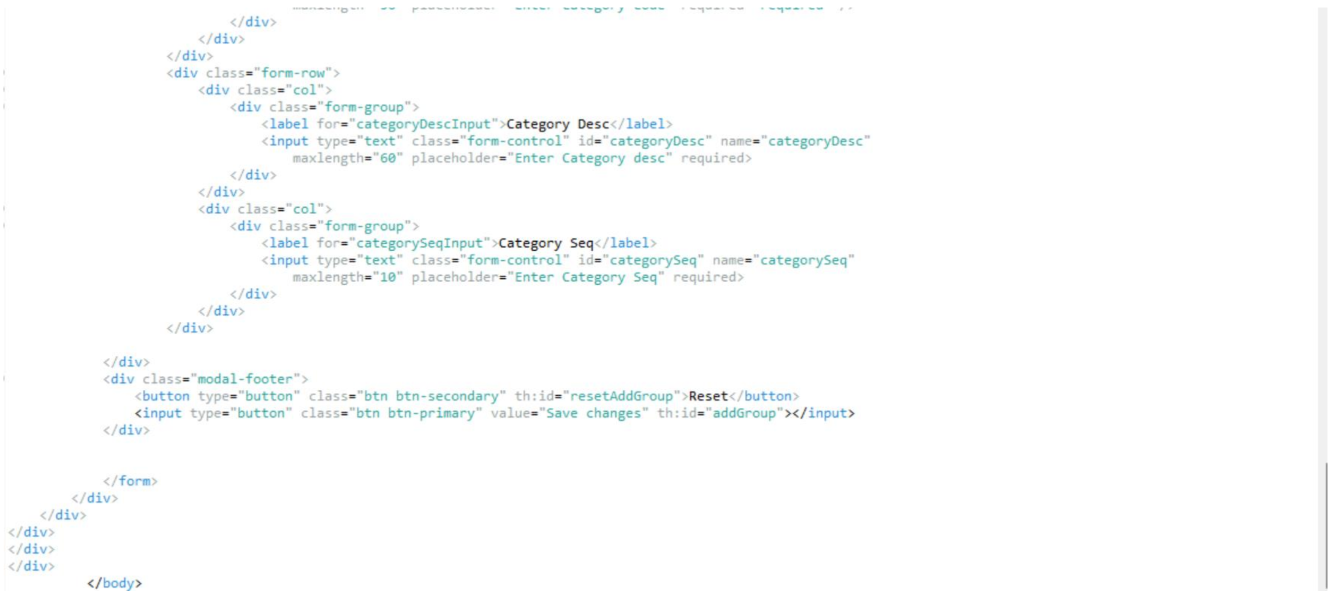
```
    }
    else if ($.trim($('#categorySeq').val())=="")
    {
    $('#addGroupMessage').html('<div class="alert de-alert-failure alert-dismissible fade show" role="alert"><strong>Category Sequence is  required</
    }
else
    {
    $('#addGroupMessage').text("");

      $("#addGroup").attr("disabled", true);
     // showSpinner
     $('#spinnerModal').appendTo("body").modal('show');
    $.post("addGroup", $("#groupForm").serialize() ,function(response)
             {
      console.log("Response------>" +JSON.stringify(response));

      $("#addGroup").attr("disabled", false);

      if(response=="Success")
          {
         // console.log("inside success");
        $('#addGroupMessage').html('<div class="alert de-alert-success alert-dismissible fade show" role="alert"><strong>Group With Category Added su
         if(btnId=="addEditGrouplink")
             {
                console.log(btnId);
                // CS 8-30-2019 - causes double submit
               // $('#editUserDataTable').DataTable().ajax.reload();

             }

          }
      else
          {
            $('#addGroupMessage').html('<div class="alert de-alert-failure alert-dismissible fade show" role="alert"><strong>' + response + '</strong
          }
      //hideSpinner
```



```
                          {
                            $('#addGroupMessage').html('<div class="alert de-alert-failure alert-dismissible fade show" role="alert"><strong>' + response + '</strong
                          }
                      //hideSpinner
                     $('#spinnerModal').appendTo("body").modal('hide');
                            $('#editGroupDataTable').DataTable().ajax.reload();
                             });
                    }



                  });


             $('#resetAddGroup').click(function()
                     {
                 $("#groupForm")[0].reset();
                    $('#addGroupMessage').text("");
                     });



        });


    </script>
<div id="addGroupModal" class="modal" tabindex="-1" role="dialog">
    <div class="modal-dialog" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title">Add Group</h5>
                <button type="button" class="close" data-dismiss="modal" aria-label="Close">
                    <span aria-hidden="true">&times;</span>
                </button>
            </div>
```

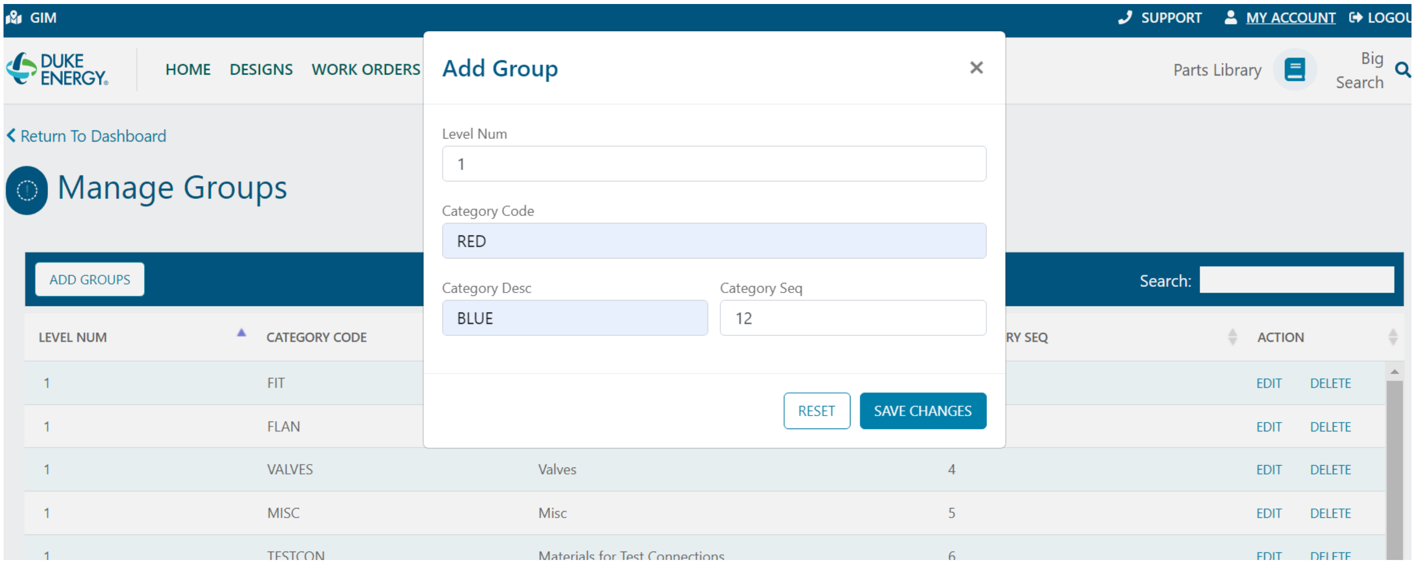


```
    </script>
<div id="addGroupModal" class="modal" tabindex="-1" role="dialog">
    <div class="modal-dialog" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title">Add Group</h5>
                <button type="button" class="close" data-dismiss="modal" aria-label="Close">
                    <span aria-hidden="true">&times;</span>
                </button>
            </div>

            <div class="modal-body">
                <form id="groupForm" name="groupForm" th:object="${partsGroupFormDTO}" method="post">
                    <div th:id="addGroupMessage"></div>

                    <div class="form-row">
                        <div class="col">
                            <div class="form-group">
                                <label for="levelNumInput">Level Num</label>
                                <input type="text" class="form-control" id="levelNum" name="levelNum" maxlength="10"
                                    placeholder="Enter Level Num" required />
                            </div>

                            <div class="form-group">
                                <label for="categoryCodeInput">Category Code</label>
                                <input type="text" class="form-control" id="categoryCode" name="categoryCode"
                                    maxlength="50" placeholder="Enter Category Code" required="required" />
                            </div>
                        </div>
                    </div>
                    <div class="form-row">
                        <div class="col">
                            <div class="form-group">
                                <label for="categoryDescInput">Category Desc</label>
                                <input type="text" class="form-control" id="categoryDesc" name="categoryDesc"
```



```
                                    maxlength="30" placeholder="Enter Category Code" required="required" />
                            </div>
                        </div>
                    </div>
                    <div class="form-row">
                        <div class="col">
                            <div class="form-group">
                                <label for="categoryDescInput">Category Desc</label>
                                <input type="text" class="form-control" id="categoryDesc" name="categoryDesc"
                                    maxlength="60" placeholder="Enter Category desc" required>
                            </div>
                        </div>
                        <div class="col">
                            <div class="form-group">
                                <label for="categorySeqInput">Category Seq</label>
                                <input type="text" class="form-control" id="categorySeq" name="categorySeq"
                                    maxlength="10" placeholder="Enter Category Seq" required>
                            </div>
                        </div>
                    </div>

            </div>
            <div class="modal-footer">
                <button type="button" class="btn btn-secondary" th:id="resetAddGroup">Reset</button>
                <input type="button" class="btn btn-primary" value="Save changes" th:id="addGroup"></input>
            </div>


            </form>
        </div>
    </div>
</div>
</div>
</div>
          </body>
```

- So here when we click on ADD GROUPS button then this modal popup will come with having these fileds.



Add Group

Level Num: 1

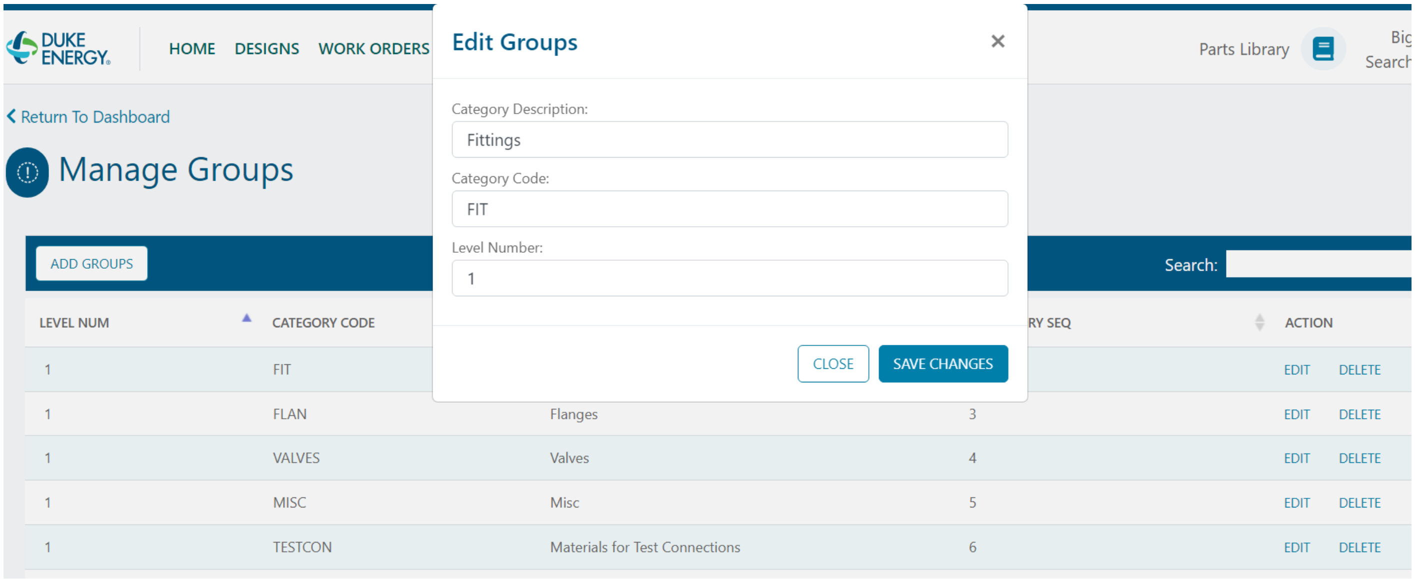
Category Code: RED

Category Desc: BLUE

Category Seq: 12

LEVEL NUM	CATEGORY CODE		RY SEQ	ACTION	
1	FIT			EDIT	DELETE
1	FLAN			EDIT	DELETE
1	VALVES	Valves	4	EDIT	DELETE
1	MISC	Misc	5	EDIT	DELETE
1	TESTCON	Materials for Test Connections	6	EDIT	DELETE

- This is the Edit modal popup with having these fields so here when user click on Edit button it will populate the data based on the fields as shown in screenshot:



Manage Groups

Edit Groups

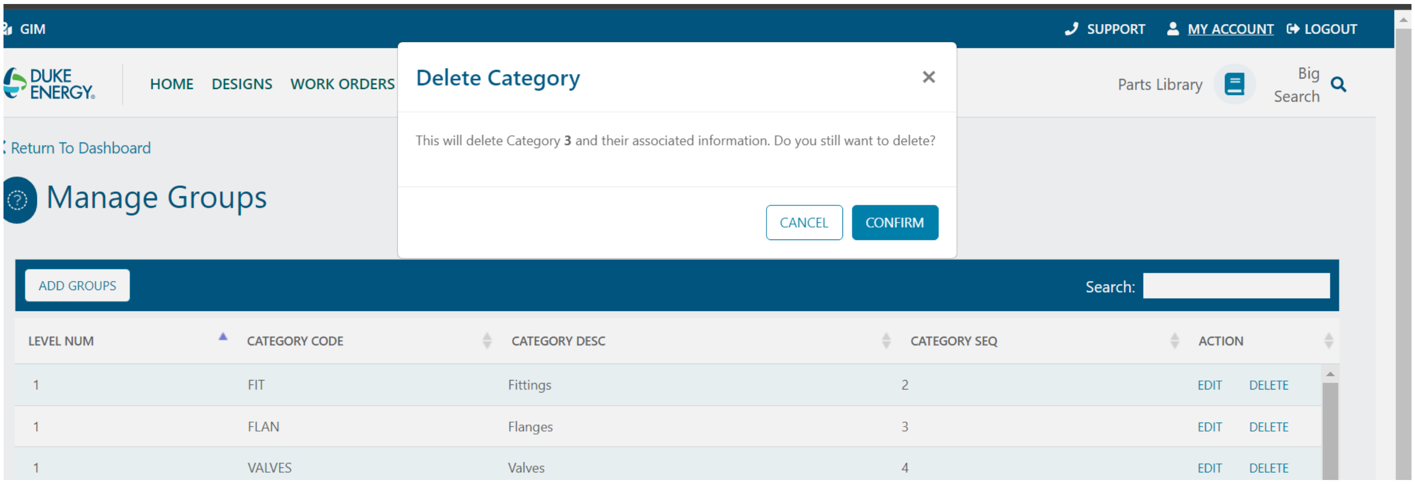
Category Description: Fittings

Category Code: FIT

Level Number: 1

LEVEL NUM	CATEGORY CODE		RY SEQ	ACTION
1	FIT			EDIT DELETE
1	FLAN	Flanges	3	EDIT DELETE
1	VALVES	Valves	4	EDIT DELETE
1	MISC	Misc	5	EDIT DELETE
1	TESTCON	Materials for Test Connections	6	EDIT DELETE

- As this is the delete popup will come when user click on delete button.

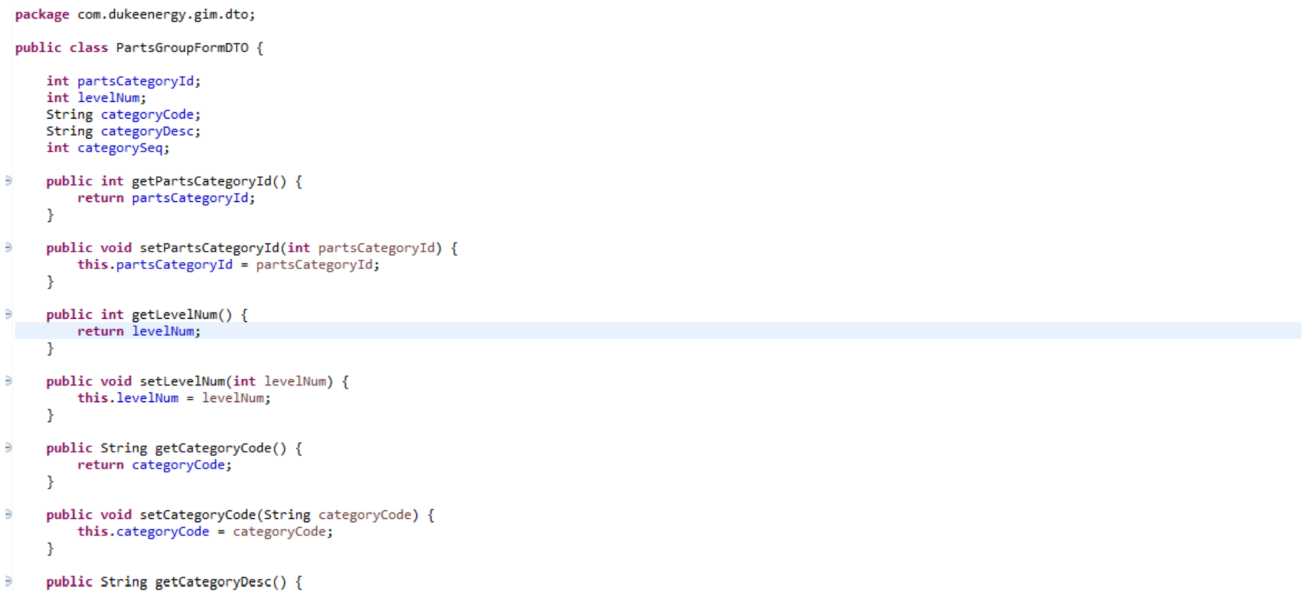


Delete Category

This will delete Category **3** and their associated information. Do you still want to delete?

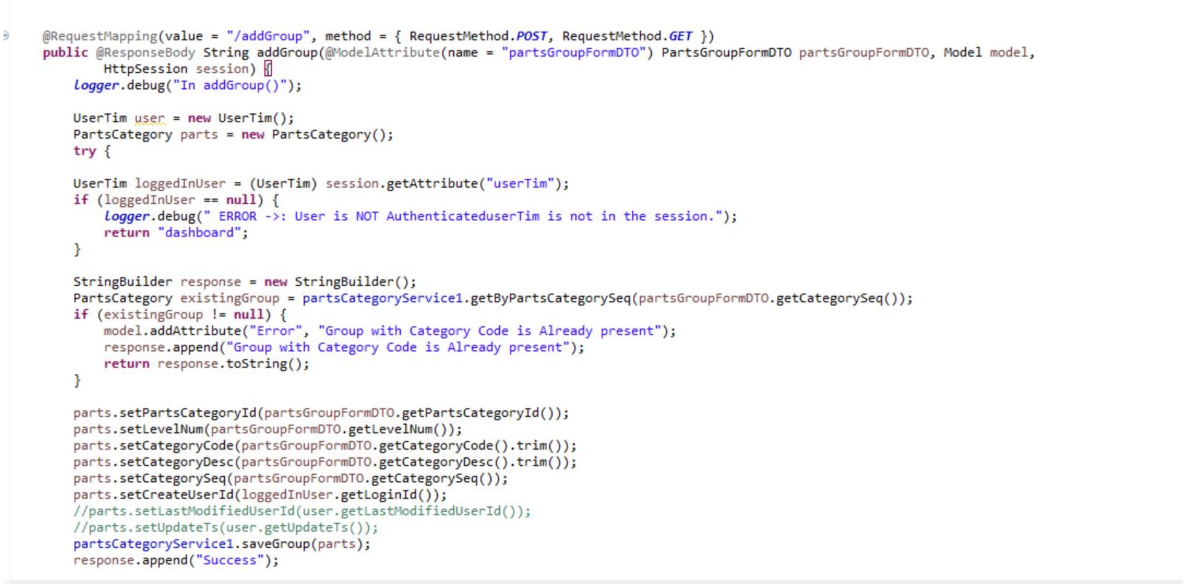
LEVEL NUM	CATEGORY CODE	CATEGORY DESC	CATEGORY SEQ	ACTION
1	FIT	Fittings	2	EDIT DELETE
1	FLAN	Flanges	3	EDIT DELETE
1	VALVES	Valves	4	EDIT DELETE

- This is class for PartsGroupFormDTO I have created which will return the response to the server with having these fields only as shown below:



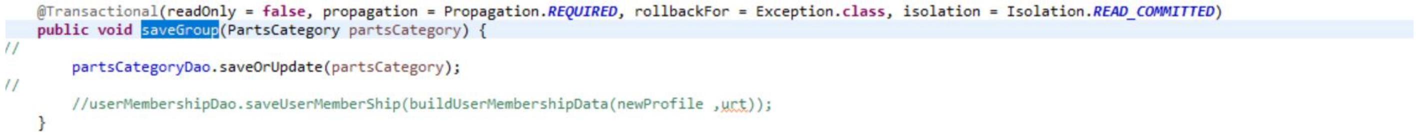
```
package com.dukeenergy.gim.dto;

public class PartsGroupFormDTO {

    int partsCategoryId;
    int levelNum;
    String categoryCode;
    String categoryDesc;
    int categorySeq;

    public int getPartsCategoryId() {
        return partsCategoryId;
    }

    public void setPartsCategoryId(int partsCategoryId) {
        this.partsCategoryId = partsCategoryId;
    }

    public int getLevelNum() {
        return levelNum;
    }

    public void setLevelNum(int levelNum) {
        this.levelNum = levelNum;
    }

    public String getCategoryCode() {
        return categoryCode;
    }

    public void setCategoryCode(String categoryCode) {
        this.categoryCode = categoryCode;
    }

    public String getCategoryDesc() {
```

- SO here when user enter all the details on the add Group popup modal so when user hit on save button then this controller /addGroup will trigger which will add the groups details into the datatabase as shown below:



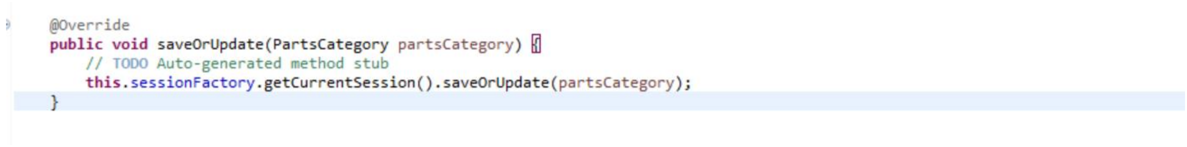
```java
@RequestMapping(value = "/addGroup", method = { RequestMethod.POST, RequestMethod.GET })
public @ResponseBody String addGroup(@ModelAttribute(name = "partsGroupFormDTO") PartsGroupFormDTO partsGroupFormDTO, Model model,
        HttpSession session) {
    logger.debug("In addGroup()");

    UserTim user = new UserTim();
    PartsCategory parts = new PartsCategory();
    try {

    UserTim loggedInUser = (UserTim) session.getAttribute("userTim");
    if (loggedInUser == null) {
        logger.debug(" ERROR ->: User is NOT AuthenticateduserTim is not in the session.");
        return "dashboard";
    }

    StringBuilder response = new StringBuilder();
    PartsCategory existingGroup = partsCategoryService1.getByPartsCategorySeq(partsGroupFormDTO.getCategorySeq());
    if (existingGroup != null) {
        model.addAttribute("Error", "Group with Category Code is Already present");
        response.append("Group with Category Code is Already present");
        return response.toString();
    }

    parts.setPartsCategoryId(partsGroupFormDTO.getPartsCategoryId());
    parts.setLevelNum(partsGroupFormDTO.getLevelNum());
    parts.setCategoryCode(partsGroupFormDTO.getCategoryCode().trim());
    parts.setCategoryDesc(partsGroupFormDTO.getCategoryDesc().trim());
    parts.setCategorySeq(partsGroupFormDTO.getCategorySeq());
    parts.setCreateUserId(loggedInUser.getLoginId());
    //parts.setLastModifiedUserId(user.getLastModifiedUserId());
    //parts.setUpdateTs(user.getUpdateTs());
    partsCategoryService1.saveGroup(parts);
    response.append("Success");
```

- This is the method I have created in the service class as saveGroup() where I have passed an PartsCategory object itself which will save the all the groups data into the table based on the partsCategoryId which is autogenerated as shown below:



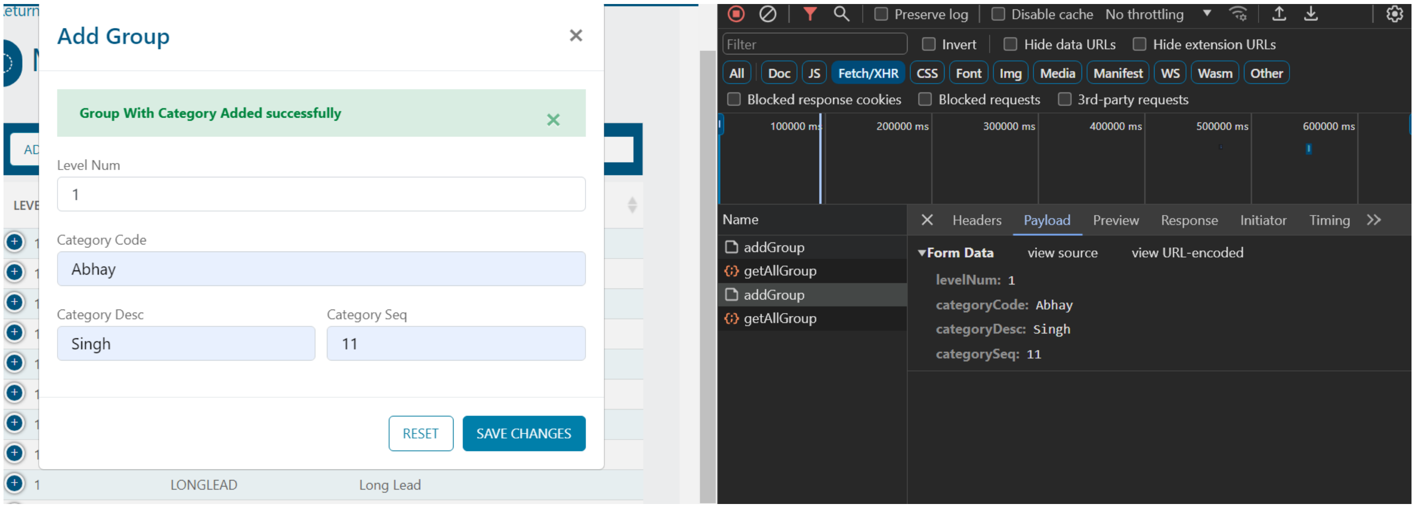
```
    @Transactional(readOnly = false, propagation = Propagation.REQUIRED, rollbackFor = Exception.class, isolation = Isolation.READ_COMMITTED)
    public void saveGroup(PartsCategory partsCategory) {
//
        partsCategoryDao.saveOrUpdate(partsCategory);
//
        //userMembershipDao.saveUserMemberShip(buildUserMembershipData(newProfile ,urt));
    }
```

- This is the method I have created saveorUpdate() in the serviceImpl class or either you can create in DAOImpl class in order to save or update the data into the datatabse with hibernate sessionfactory



```
@Override
public void saveOrUpdate(PartsCategory partsCategory) {
    // TODO Auto-generated method stub
    this.sessionFactory.getCurrentSession().saveOrUpdate(partsCategory);
}
```

- So here when the group data inserted successfully inside the database it will return the message as shown in the screenshot and beside this in the payload we can see that these data is getting passed from the partGroupFormDTO class to the server as shown in the attached screenshot:



Add Group

Group With Category Added successfully

Level Num: 1

Category Code: Abhay

Category Desc: Singh

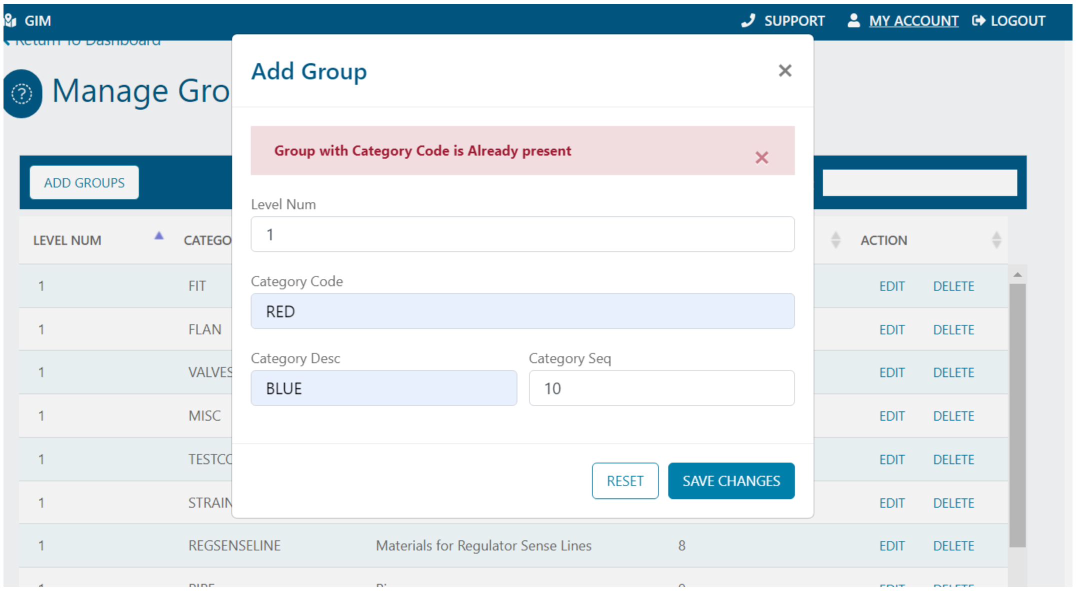
Category Seq: 11

LONGLEAD Long Lead

Form Data

```
levelNum: 1
categoryCode: Abhay
categoryDesc: Singh
categorySeq: 11
```

- So here when user hit multiple time on the save button so if the entered data is already present inside the datatabse so in the controller it will check as partsCategoryId if it is not null then it means the same data is already present then I have append the response as (Group with Category Code is Already present) as shown below:



Group with Category Code is Already present

Level Num: 1

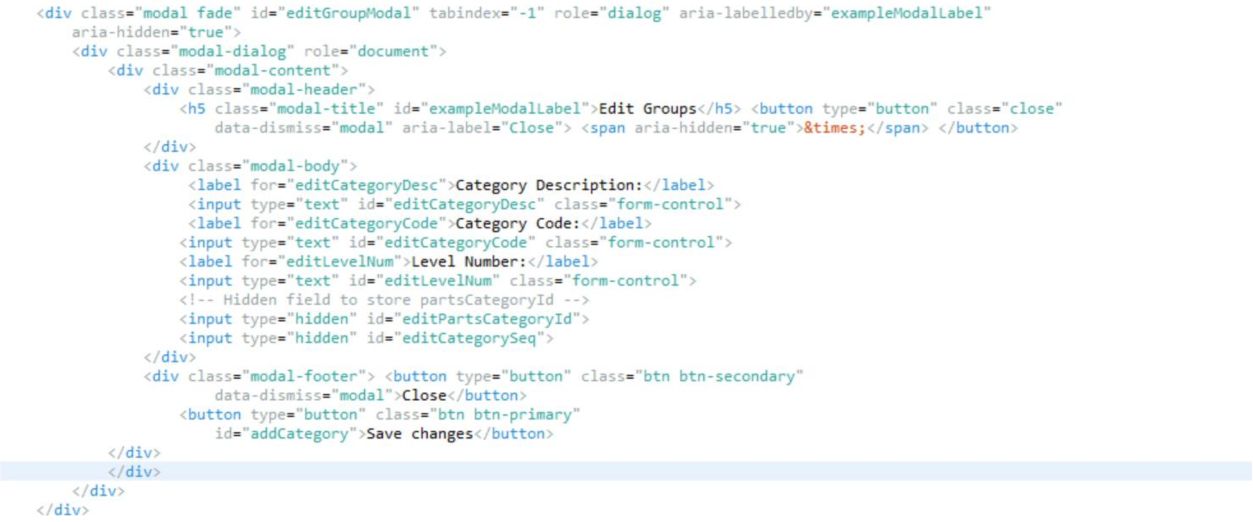
Category Code: RED

Category Desc: BLUE

Category Seq: 10

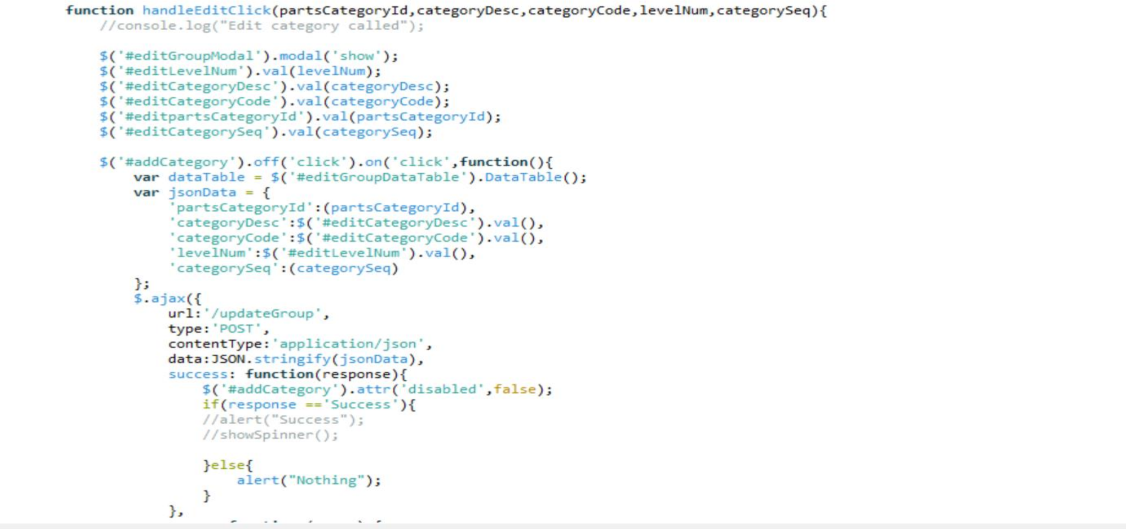
LEVEL NUM	CATEGO			ACTION	
1	FIT			EDIT	DELETE
1	FLAN			EDIT	DELETE
1	VALVES			EDIT	DELETE
1	MISC			EDIT	DELETE
1	TESTCO			EDIT	DELETE
1	STRAIN			EDIT	DELETE
1	REGSENSELINE	Materials for Regulator Sense Lines	8	EDIT	DELETE

- This is the editGroup modal code.



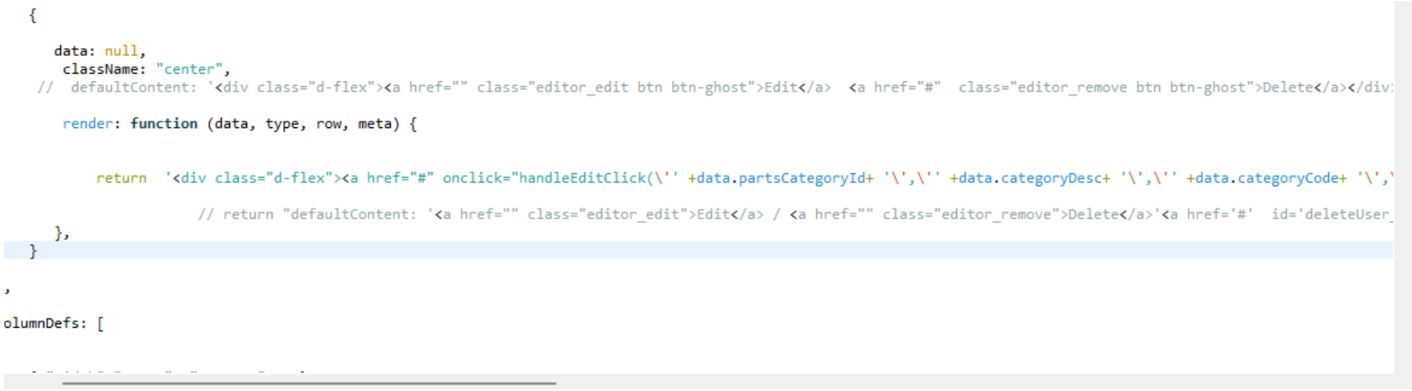
```
<div class="modal fade" id="editGroupModal" tabindex="-1" role="dialog" aria-labelledby="exampleModalLabel"
    aria-hidden="true">
    <div class="modal-dialog" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title" id="exampleModalLabel">Edit Groups</h5> <button type="button" class="close"
                    data-dismiss="modal" aria-label="Close"> <span aria-hidden="true">&times;</span> </button>
            </div>
            <div class="modal-body">
                 <label for="editCategoryDesc">Category Description:</label>
                 <input type="text" id="editCategoryDesc" class="form-control">
                 <label for="editCategoryCode">Category Code:</label>
                <input type="text" id="editCategoryCode" class="form-control">
                <label for="editLevelNum">Level Number:</label>
                <input type="text" id="editLevelNum" class="form-control">
                <!-- Hidden field to store partsCategoryId -->
                <input type="hidden" id="editPartsCategoryId">
                <input type="hidden" id="editCategorySeq">
            </div>
            <div class="modal-footer"> <button type="button" class="btn btn-secondary"
                    data-dismiss="modal">Close</button>
                <button type="button" class="btn btn-primary"
                    id="addCategory">Save changes</button>
        </div>
        </div>
    </div>
</div>
```

- This is the function getting called when user hit on Edit button on the data table I have passed four parameters inside the function so when this function is called it will show the modal popup and based on the id’s defined for an particular fields it will append that value to the field and finally when user edit the details then when user click on save changes then this add Category method is getting called and here I am passing this jsonData to the controller method based on the edited field value by the user.



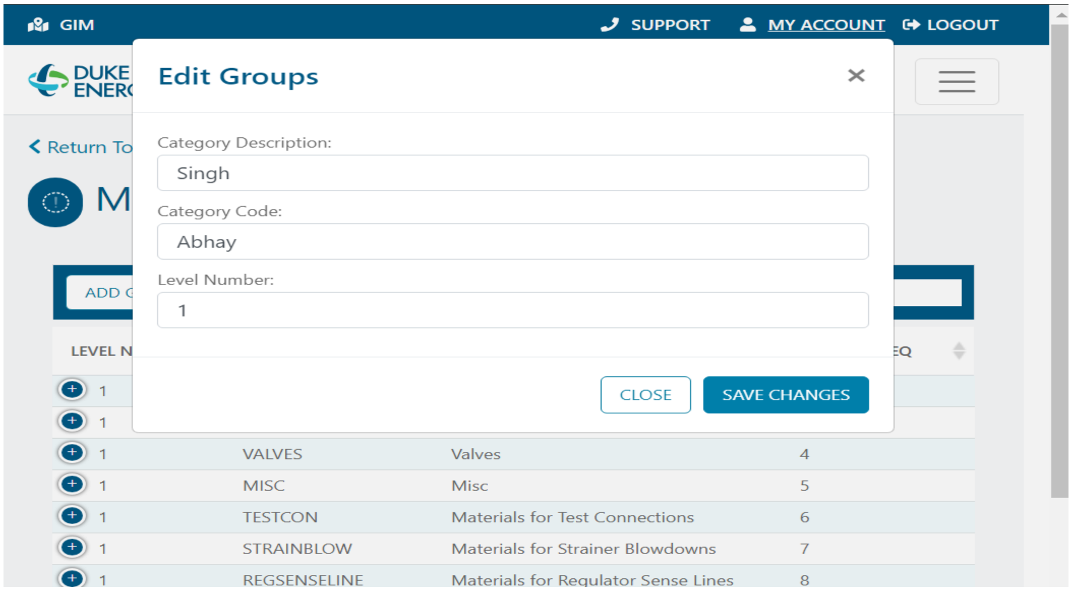
```
function handleEditClick(partsCategoryId,categoryDesc,categoryCode,levelNum,categorySeq){
    //console.log("Edit category called");

    $('#editGroupModal').modal('show');
    $('#editLevelNum').val(levelNum);
    $('#editCategoryDesc').val(categoryDesc);
    $('#editCategoryCode').val(categoryCode);
    $('#editpartsCategoryId').val(partsCategoryId);
    $('#editCategorySeq').val(categorySeq);

    $('#addCategory').off('click').on('click',function(){
        var dataTable = $('#editGroupDataTable').DataTable();
        var jsonData = {
            'partsCategoryId':(partsCategoryId),
            'categoryDesc':$('#editCategoryDesc').val(),
            'categoryCode':$('#editCategoryCode').val(),
            'levelNum':$('#editLevelNum').val(),
            'categorySeq':(categorySeq)
        };
        $.ajax({
            url:'/updateGroup',
            type:'POST',
            contentType:'application/json',
            data:JSON.stringify(jsonData),
            success: function(response){
                $('#addCategory').attr('disabled',false);
                if(response =='Success'){
                //alert("Success");
                //showSpinner();

                }else{
                    alert("Nothing");
                }
            },
```

- Here I am passing these parameters while calling the handleEditClick function (here I am passing the parameter values which is getting from data as shown below:)



```
  {

     data: null,
      className: "center",
   //  defaultContent: '<div class="d-flex"><a href="" class="editor_edit btn btn-ghost">Edit</a>  <a href="#"  class="editor_remove btn btn-ghost">Delete</a></div

      render: function (data, type, row, meta) {


          return  '<div class="d-flex"><a href="#" onclick="handleEditClick(\'' +data.partsCategoryId+ '\',\'' +data.categoryDesc+ '\',\'' +data.categoryCode+ '\','

                      // return "defaultContent: '<a href="" class="editor_edit">Edit</a> / <a href="" class="editor_remove">Delete</a>'<a href='#'  id='deleteUser_
     },
  }

,

olumnDefs: [
```

- Here we can saw the edit values inside the modal popup.



Edit Groups

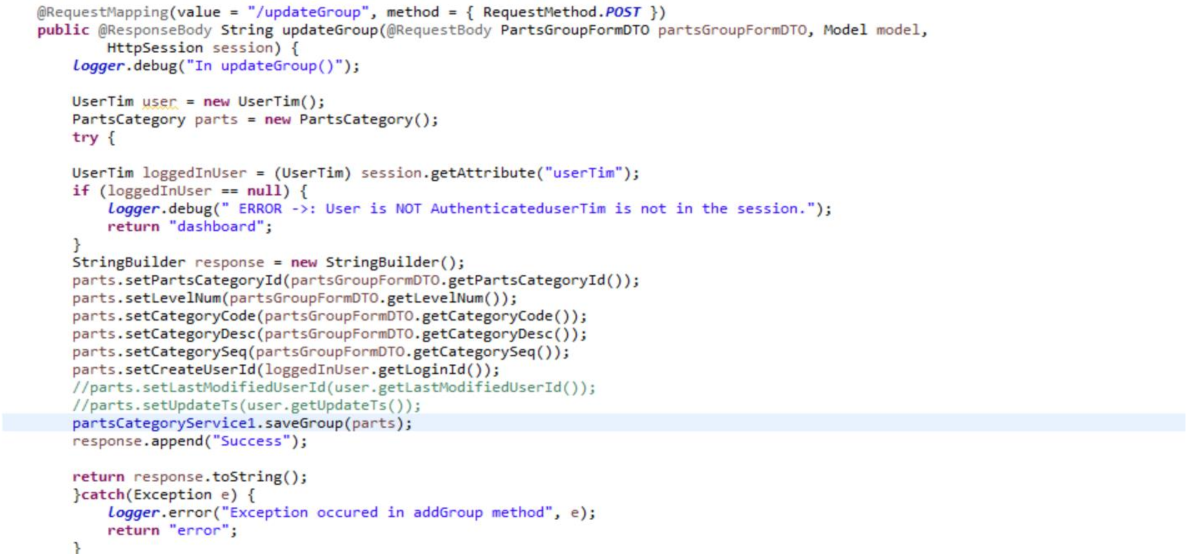
Category Description: Singh

Category Code: Abhay

Level Number: 1

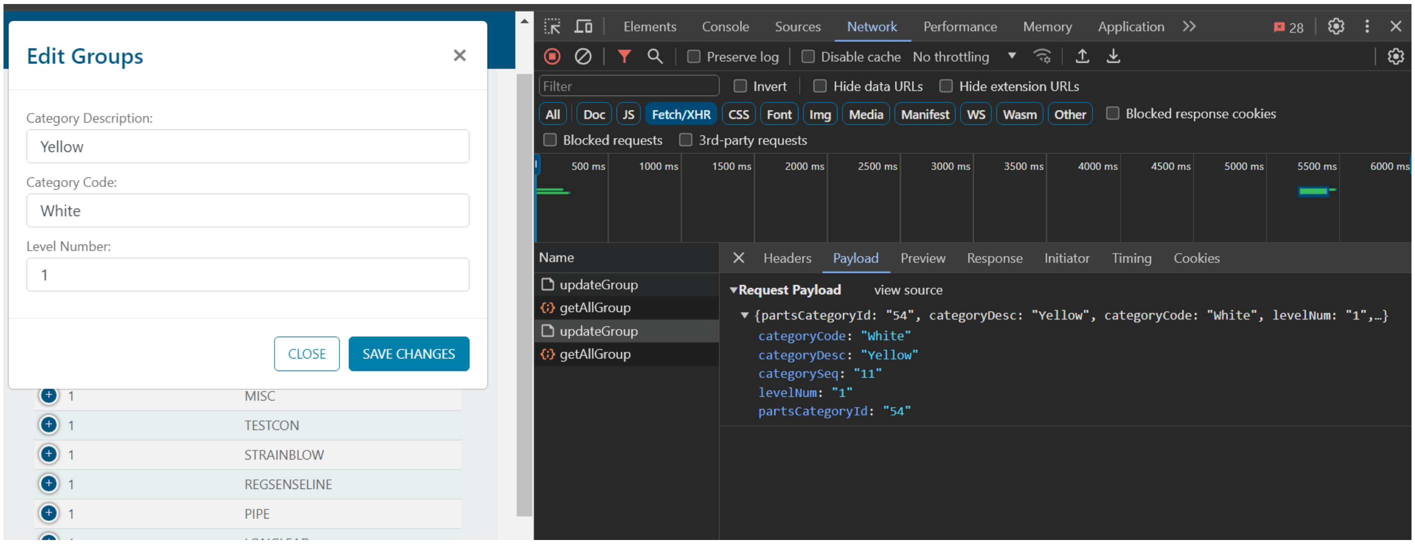
1			
1			
1	VALVES	Valves	4
1	MISC	Misc	5
1	TESTCON	Materials for Test Connections	6
1	STRAINBLOW	Materials for Strainer Blowdowns	7
1	REGSENSELINE	Materials for Regulator Sense Lines	8

- Here this is the controller method is getting invoked when user hit on saveChanges button while doing an Edit so here I am passing jsondata which is getting from edited fields while doing an Edit from modal to here which is getting stored inside the partsGroupFormDTO object based on the filed which is defined inside the PartsGroupFormDTO and after that it will update the datatabase as shown below:



```
@RequestMapping(value = "/updateGroup", method = { RequestMethod.POST })
public @ResponseBody String updateGroup(@RequestBody PartsGroupFormDTO partsGroupFormDTO, Model model,
        HttpSession session) {
    logger.debug("In updateGroup()");

    UserTim user = new UserTim();
    PartsCategory parts = new PartsCategory();
    try {

    UserTim loggedInUser = (UserTim) session.getAttribute("userTim");
    if (loggedInUser == null) {
        logger.debug(" ERROR ->: User is NOT AuthenticateduserTim is not in the session.");
        return "dashboard";
    }
    StringBuilder response = new StringBuilder();
    parts.setPartsCategoryId(partsGroupFormDTO.getPartsCategoryId());
    parts.setLevelNum(partsGroupFormDTO.getLevelNum());
    parts.setCategoryCode(partsGroupFormDTO.getCategoryCode());
    parts.setCategoryDesc(partsGroupFormDTO.getCategoryDesc());
    parts.setCategorySeq(partsGroupFormDTO.getCategorySeq());
    parts.setCreateUserId(loggedInUser.getLoginId());
    //parts.setLastModifiedUserId(user.getLastModifiedUserId());
    //parts.setUpdateTs(user.getUpdateTs());
    partsCategoryService1.saveGroup(parts);
    response.append("Success");

    return response.toString();
    }catch(Exception e) {
        logger.error("Exception occured in addGroup method", e);
        return "error";
    }
```

- So here when I hit on save Changes button then in the payload you can find json data is getting passed which I passed inside the method handleEditClick()
  and this data as json response is getting passed to the controller method as shown below:



Edit Groups

Category Description: Yellow

Category Code: White

Level Number: 1

1	MISC
1	TESTCON
1	STRAINBLOW
1	REGSENSELINE
1	PIPE

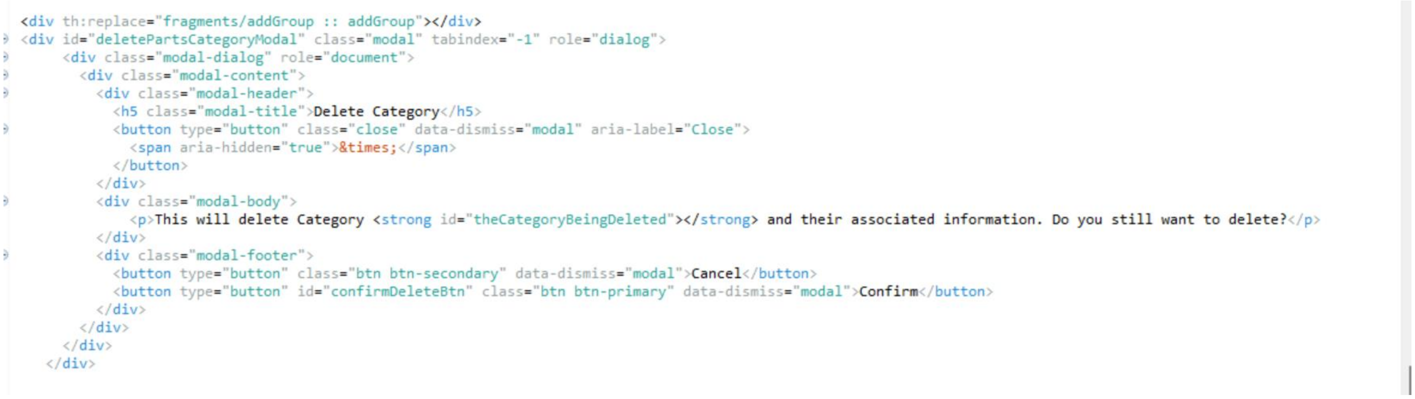
Name

updateGroup
getAllGroup
updateGroup
getAllGroup

Request Payload

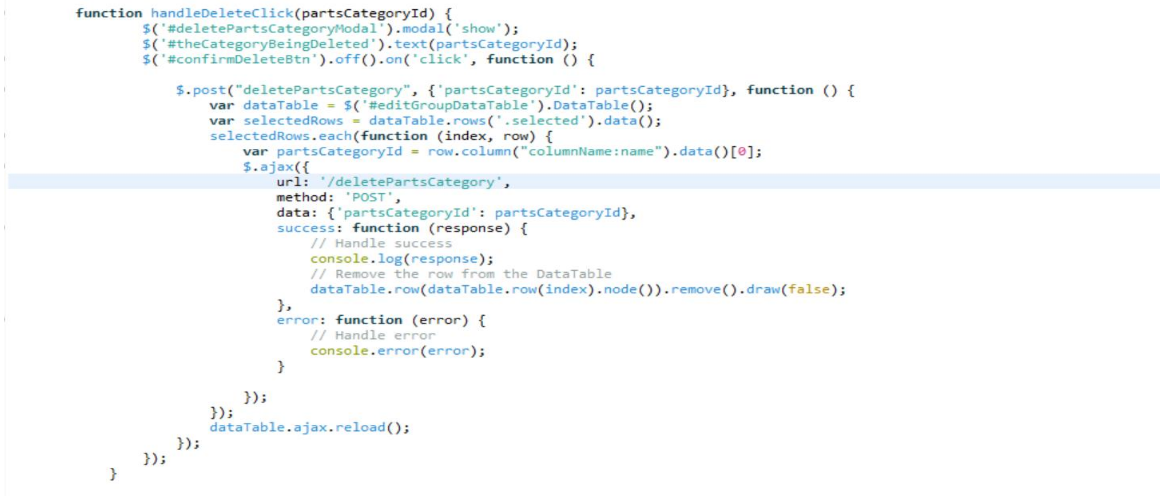
```
▼ {partsCategoryId: "54", categoryDesc: "Yellow", categoryCode: "White", levelNum: "1",…}
    categoryCode: "White"
    categoryDesc: "Yellow"
    categorySeq: "11"
    levelNum: "1"
    partsCategoryId: "54"
```

- Here this is the delete modal popup as shown below:



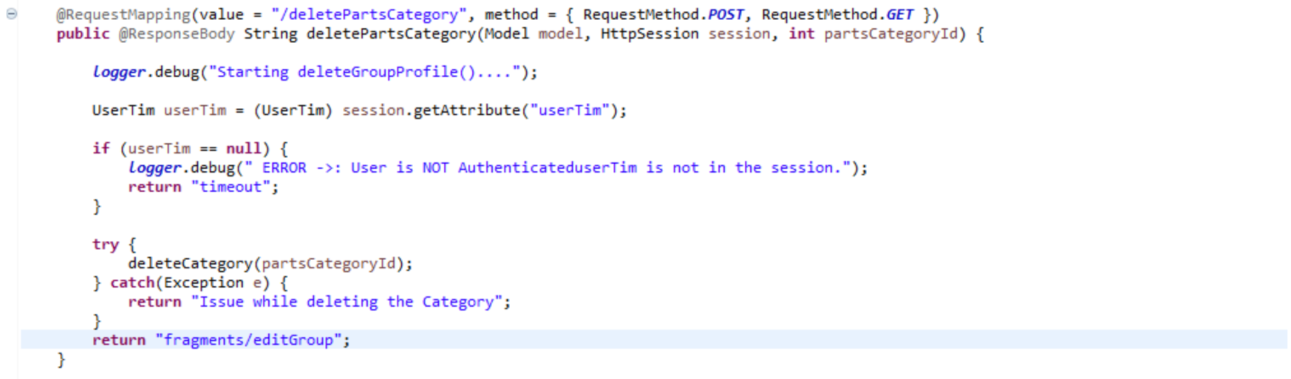
```
<div th:replace="fragments/addGroup :: addGroup"></div>
<div id="deletePartsCategoryModal" class="modal" tabindex="-1" role="dialog">
     <div class="modal-dialog" role="document">
       <div class="modal-content">
         <div class="modal-header">
           <h5 class="modal-title">Delete Category</h5>
           <button type="button" class="close" data-dismiss="modal" aria-label="Close">
             <span aria-hidden="true">&times;</span>
           </button>
         </div>
         <div class="modal-body">
             <p>This will delete Category <strong id="theCategoryBeingDeleted"></strong> and their associated information. Do you still want to delete?</p>
         </div>
         <div class="modal-footer">
           <button type="button" class="btn btn-secondary" data-dismiss="modal">Cancel</button>
           <button type="button" id="confirmDeleteBtn" class="btn btn-primary" data-dismiss="modal">Confirm</button>
         </div>
       </div>
     </div>
   </div>
```

- Here this function is getting called when user hit on delete button and it will show the delete modal popup and based on the field id’s it will print the text value on the popup i.e partsCategoryId and when user hit on confirm button then it will return inside the confirmDeleteBtn function then using AJAX post call it will trigger the /deletePartsCategory controller endpoint and there I have passed an partsCategoryId as shown below:



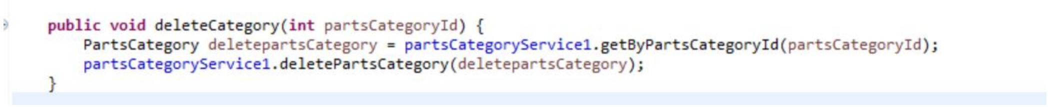
```
function handleDeleteClick(partsCategoryId) {
        $('#deletePartsCategoryModal').modal('show');
        $('#theCategoryBeingDeleted').text(partsCategoryId);
        $('#confirmDeleteBtn').off().on('click', function () {

            $.post("deletePartsCategory", {'partsCategoryId': partsCategoryId}, function () {
                var dataTable = $('#editGroupDataTable').DataTable();
                var selectedRows = dataTable.rows('.selected').data();
                selectedRows.each(function (index, row) {
                    var partsCategoryId = row.column("columnName:name").data()[0];
                    $.ajax({
                        url: '/deletePartsCategory',
                        method: 'POST',
                        data: {'partsCategoryId': partsCategoryId},
                        success: function (response) {
                            // Handle success
                            console.log(response);
                            // Remove the row from the DataTable
                            dataTable.row(dataTable.row(index).node()).remove().draw(false);
                        },
                        error: function (error) {
                            // Handle error
                            console.error(error);
                        }

                    });
                });
                dataTable.ajax.reload();
            });
        });
    }
```

- Here it will trigger this controller method and it will invoke the method deleteCategory() from the try block based on the partsCategoryId which is getting passed from the froentend code as shown below:



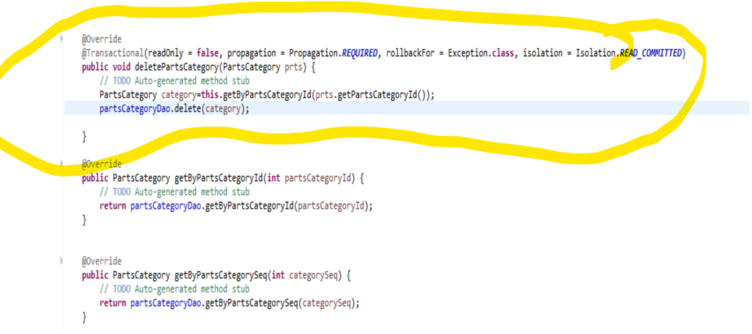
```
@RequestMapping(value = "/deletePartsCategory", method = { RequestMethod.POST, RequestMethod.GET })
public @ResponseBody String deletePartsCategory(Model model, HttpSession session, int partsCategoryId) {

    logger.debug("Starting deleteGroupProfile()....");

    UserTim userTim = (UserTim) session.getAttribute("userTim");

    if (userTim == null) {
        logger.debug(" ERROR ->: User is NOT AuthenticateduserTim is not in the session.");
        return "timeout";
    }

    try {
        deleteCategory(partsCategoryId);
    } catch(Exception e) {
        return "Issue while deleting the Category";
    }
    return "fragments/editGroup";
}
```

- Here I have created deleteCategory method inside the service class which will get the data based on partsCategoryId as shown below:

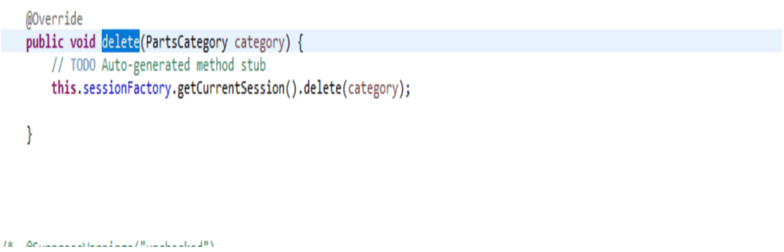


```java
public void deleteCategory(int partsCategoryId) {
    PartsCategory deletepartsCategory = partsCategoryService1.getByPartsCategoryId(partsCategoryId);
    partsCategoryService1.deletePartsCategory(deletepartsCategory);
}
```

- SO here I have created an delete method inside the serviceImpl class in order to delete the data based on the partsCategoryId.

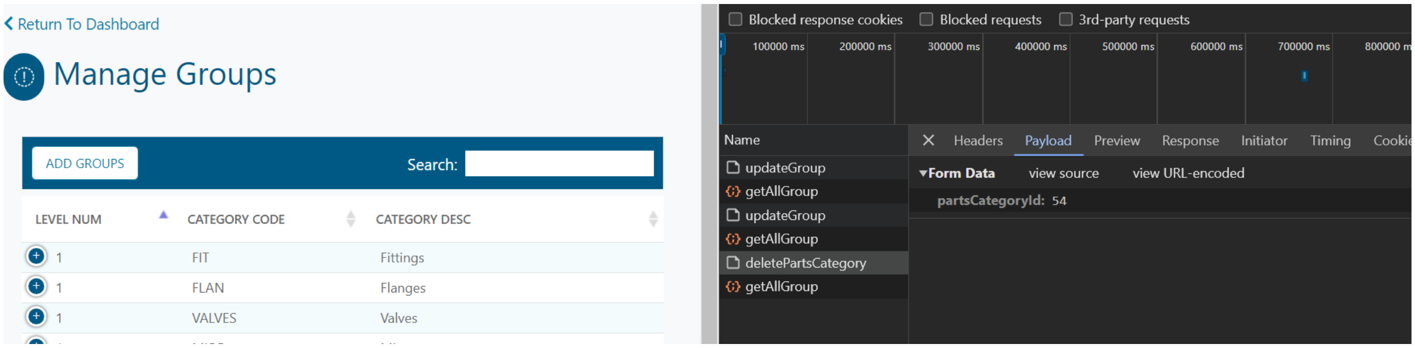


```
@Override
@Transactional(readOnly = false, propagation = Propagation.REQUIRED, rollbackFor = Exception.class, isolation = Isolation.READ_COMMITTED)
public void deletePartsCategory(PartsCategory prts) {
    // TODO Auto-generated method stub
    PartsCategory category=this.getByPartsCategoryId(prts.getPartsCategoryId());
    partsCategoryDao.delete(category);

}

@Override
public PartsCategory getByPartsCategoryId(int partsCategoryId) {
    // TODO Auto-generated method stub
    return partsCategoryDao.getByPartsCategoryId(partsCategoryId);
}


@Override
public PartsCategory getByPartsCategorySeq(int categorySeq) {
    // TODO Auto-generated method stub
    return partsCategoryDao.getByPartsCategorySeq(categorySeq);
}
```



```
    @Override
    public void delete(PartsCategory category) {
        // TODO Auto-generated method stub
        this.sessionFactory.getCurrentSession().delete(category);

    }



/*  @SuppressWarnings("unchecked")
```

- Here we find after deletion happens successfully we can find from the payload we can find partsCategoryId is getting passed based on this it is getting passed inside the controller method as shown below:



< Return To Dashboard

Manage Groups

LEVEL NUM	CATEGORY CODE	CATEGORY DESC
1	FIT	Fittings
1	FLAN	Flanges
1	VALVES	Valves

Name
updateGroup
getAllGroup
updateGroup
getAllGroup
deletePartsCategory
getAllGroup

▾Form Data view source view URL-encoded

partsCategoryId: 54