function test()

{

var data = {

"items": [{

"id": 1, "category": "cat1"

}, {

"id": 2, "category": "cat2"

}, {

"id": 3, "category": "cat1"

}]

};

var returnedData = $.grep(data.items, function (element, index) {

return element.id == 2;

});

alert(returnedData[0].id + " " + returnedData[0].category);

}

Because id==2 it displays 2 , cat2

Si encuentra grupo rol devuelve true, sino false

$rootScope.isInRole = function(roleGroupID)

{

var ret = false;

if (\_employeeObject != null &&

\_employeeObject.EmpRoles != null &&

\_employeeObject.EmpRoles.length > 0)

{

ret = (jQuery.grep(\_employeeObject.EmpRoles, function (role) {

return role.RoleGroup == roleGroupID

}).length > 0);

}

return ret;

}

Usage:

$scope.isSuperUser = function () {

alert($rootScope.isInRole(2));

return false;

};

I created following two new (base) classes for the existing EmployeeEnity and RoleEntity

public class EmployeeMini

    {

        public string EmployeeID { get; set; }

        public string FirstName { get; set; }

        public string LastName { get; set; }

        public IList<RoleEntityMini> Roles { get; set; }

    }

public class RoleEntityMini

    {

        public Constant.RoleGroup RoleGroup { get; set; }

        public string RoleGroupName { get; set; }

    }

i.e moved the required JSON properties from original classes to base classes and now inheriting EmployeeEntity and RoleEntity from the respective base classes.

Created a new handler wfmEmpRoleInfo.ashx (found in application root folder) which copies employee properties from session and populates “employeeMiniObject” and Parses it into following JSON string.

var \_employeeObject = {"Roles":[

{"RoleGroup":1,"RoleGroupName":"User"},

{"RoleGroup":2,"RoleGroupName":"Super User"},

{"RoleGroup":3,"RoleGroupName":"LOB Admin"},

{"RoleGroup":4,"RoleGroupName":"Metric Admin"},

{"RoleGroup":5,"RoleGroupName":"CCA Role"},

{"RoleGroup":6,"RoleGroupName":"WFM User"},

{"RoleGroup":7,"RoleGroupName":"Program Mgmnt Admin"}],

"EmployeeID":"100140255","FirstName":"Amit","LastName":"Bhatnagar"}

This handler behaves like a external javascript files which provides global \_employeeObject javascript variable

<script type="text/javascript" src="/wfmEmpRoleInfo.ashx"></script>

Challenges:

each time we do full page refresh the \_employeeObject will be refreshed, but when page not doing full refresh from long time due to Angular/Ajax functionality,need to find some way to refresh the \_employeeObject Periodically to reflect the role changes happened in database.

\*\*Can we use SignalR in this case?

\*\*Also should be able to detect (server) session timeouts to destroy this client side object (\_employeeObject)

\*\*A cunning user can change this object’s properties (can add/modify roles) through some script injecting;

Still working to add some security around this object.