```
/**
 * Created by Alain Dechorgnat on 12/13/13.
 */

var StatusApp = angular.module('StatusApp',
['D3Directives','ngCookies','ngAnimate','InkscopeCommons'])
    .filter('bytes', funcBytesFilter)
    .filter('duration', funcDurationFilter);

StatusApp.controller("statusCtrl", function ($rootScope, $scope,
$http , $cookieStore) {
    $scope.journal = [];
    $scope.osdControl =0;
    $scope.statOK = true;

    $scope.refreshPG = true;
    $scope.refreshPGcontrolClass = "icon-pause";

    $scope.viewControlPanel = false;
    $scope.viewMonitorModule = testAndSetCookie
('viewMonitorModule',true);
    $scope.viewCapacityModule = testAndSetCookie
('viewCapacityModule',true);
    $scope.viewPoolModule = testAndSetCookie
('viewPoolModule',true);
    $scope.viewOsdModule = testAndSetCookie
('viewOsdModule',true);
    $scope.viewPgStatusModule = testAndSetCookie
('viewPgStatusModule',true);
    $scope.viewMdsModule = testAndSetCookie
('viewMdsModule',true);

    function testAndSetCookie(param,defaultValue) {
        var value = $cookieStore.get(param);
        if (typeof value ==="undefined") {
            $cookieStore.put(param,defaultValue);
            value = defaultValue;
        }
        return value;
    }

    // instantiate our graph!
    var graph = new Rickshaw.Graph( {
        element: document.getElementById("iopschart"),
        width: 300,
        height: 30,
        renderer: 'line',
        series: new Rickshaw.Series.FixedDuration([{ name: 'read'
},{ name: 'write' },{ name: 'recovery' }], undefined, {
            timeInterval: 3000,
            maxDataPoints: 100,
            timeBase: new Date().getTime() / 1000
```

```
        })
    } );
    var hoverDetail = new Rickshaw.Graph.HoverDetail( {
        graph: graph,
        xFormatter: function(x) { return  ""; },
        yFormatter: function(y) { return  funcBytes(y)+ "/s" ;}
    } );
    var yAxis = new Rickshaw.Graph.Axis.Y({
        graph: graph,
        height: 30,
        orientation: 'left',
        tickFormat: Rickshaw.Fixtures.Number.formatKMBT,
        element: document.getElementById('y_axis')
    });
    yAxis.render();
    graph.render();

    //refresh data every x seconds
    refreshData();
    refreshPGData();
    setInterval(function () {
        refreshData()
    }, 5 * 1000);
    setInterval(function () {
        refreshPGData()
    }, 10 * 1000);


    // start refreshOSDData when fsid is available
    var waitForFsid = function ($rootScope, $http,$scope){
        typeof $rootScope.fsid !== "undefined"? startRefresh
($rootScope, $http,$scope) : setTimeout(function () {waitForFsid
($rootScope, $http,$scope)}, 1000);
        function startRefresh($rootScope, $http,$scope){
            refreshOSDData();
            setInterval(function () {
                refreshOSDData()
            }, 5 * 1000);
        }
    }
    waitForFsid($rootScope, $http,$scope);



    function refreshPGData() {
        $scope.date = new Date();
        $http({method: "get", url: cephRestApiURL +
"df.json",timeout:8000})
            .success(function (data, status) {
                $scope.nbPools = data.output.pools.length;;
            });
        $http({method: "get", url: cephRestApiURL +
```

```
"pg/dump_stuck.json",timeout:8000})
            .success(function (data, status) {
                // fetching stuck pg list
                var pg_stats = data.output;
                var pools = [];
                for (var i = 0; i < pg_stats.length; i++) {
                    var pg = pg_stats[i];
                    //console.log(pg.pgid + " : " + pg.state)
                    var numPool = pg.pgid.split(".")[0];
                    pools[numPool] = false;
                }
                var cnt = 0;
                for (var i = 0; i < pools.length; i++) {
                    if (typeof pools[i] !== "undefined") {
                        ++cnt;
                    }
                }
                $scope.uncleanPools = cnt;
                if (typeof $scope.nbPools !== "undefined")
$scope.cleanPools = $scope.nbPools -cnt;
            });
    };

    function refreshOSDData() {
        $scope.date = new Date();
        var filter = {
            "$select":{},
            "$template":{
                "stat":1
            }
        }
        $http({method: "post", url: inkscopeCtrlURL +
$rootScope.fsid+"/osd", params :{"depth":1}
,data:filter,timeout:4000})
            .success(function (data, status) {
                if (data[0].stat == null)
                    $scope.osdControl ="-";
                else
                    $scope.osdControl = ((+$scope.date)-data
[0].stat.timestamp)/1000 ;
                $scope.osdsInUp = 0;
                $scope.osdsInDown = 0;
                $scope.osdsOutUp = 0;
                $scope.osdsOutDown = 0;
                $scope.statOK = true;
                for (var i = 0; i < data.length; i++) {
                    if (!data[i].lost) {
                        if (data[i].stat ==null){
                            console.error("missing stat for
"+data[i].node.name);
                            $scope.statOK = false;
                        }
```

```
                        else {
                            if (data[i].stat.in) {
                                if (data[i].stat.up)
$scope.osdsInUp++; else $scope.osdsInDown++;
                            }
                            else {
                                if (data[i].stat.up)
$scope.osdsOutUp++; else $scope.osdsOutDown++;
                            }
                        }
                    }
                }
        })
            .error (function (data, status){
                console.error("can't fetch osd info in mongo");
                $scope.statOK = false;
    });
    };


    function refreshData() {
        //console.log("refreshing data...");
        $scope.date = new Date();
        $http({method: "get", url: cephRestApiURL +
"status.json",timeout:4000})
            .success(function (data) {
                $scope.pgmap = data.output.pgmap;
                if ( typeof  data.output.pgmap.degraded_objects
=== "undefined" ) $scope.pgmap.degraded_objects=0;


//                $scope.mdsmap = data.output.mdsmap;
//                $scope.mdsmap.up_standby = data.output.mdsmap
["up:standby"];
                if (typeof data.output.mdsmap !== "undefined"){
                    $scope.mdsmap = data.output.mdsmap;
                    $scope.mdsmap.up_standby = data.output.mdsmap
["up:standby"];
                }
                else {
                    $scope.mdsmap = data.output.fsmap;
                    $scope.mdsmap.up_standby = data.output.fsmap
["up:standby"];
                }

                $scope.percentUsed = $scope.pgmap.bytes_used /
$scope.pgmap.bytes_total;


                if ($scope.refreshPG) $scope.pgsByState =
$scope.pgmap.pgs_by_state;
```

```
                $scope.read = (data.output.pgmap.read_bytes_sec ?
data.output.pgmap.read_bytes_sec : 0);
                $scope.write = (data.output.pgmap.write_bytes_sec
? data.output.pgmap.write_bytes_sec : 0);
                $scope.recovery =
(data.output.pgmap.recovering_bytes_per_sec ?
data.output.pgmap.recovering_bytes_per_sec : 0);

                var iopsdata = { read: $scope.read , write :
$scope.write , recovery : $scope.recovery };
                graph.series.addData(iopsdata);
                yAxis.render();
                graph.render();

                $scope.health = {};
                $scope.health.severity =
data.output.health.overall_status;
                // there may be several messages under
data.output.health.summary
                $scope.health.summary="";
                var i = 0;
                while(typeof data.output.health.summary[i] !==
"undefined"){
                    if ($scope.health.summary!="")
$scope.health.summary+=" | ";
                    $scope.health.summary +=
data.output.health.summary[i].summary;
                    i++;
                }
                if ($scope.health.summary==""){
                    if (data.output.health.detail[0])
                        $scope.health.summary =
data.output.health.detail[0];
                    else
                        //remove HEALTH_ in severity
                        $scope.health.summary =
$scope.health.severity.substring(7);
                }
                $rootScope.healthSeverity =
$scope.health.severity;
                historise();

                $scope.mons = data.output.monmap.mons;

                for (var i = 0; i < $scope.mons.length; i++) {
                    var mon = $scope.mons[i];
                    mon.health = "HEALTH_UNKNOWN"; // default for
styling purpose
                    mon.quorum = "out";          // default for
styling purpose
                    for (var j = 0; j <
data.output.quorum_names.length; j++) {
```

```
                        if (mon.name == data.output.quorum_names
[j]) {
                            mon.quorum = "in";
                            break
                        }
                    }
                    if (data.output.health.timechecks.mons) //not
always defined
                        for (var j = 0; j <
data.output.health.timechecks.mons.length; j++) {
                            mon2 =
data.output.health.timechecks.mons[j];
                            if (mon.name == mon2.name) {
                                for (key in mon2) mon[key] = mon2
[key];
                                break
                            }
                        }
                    for (var j = 0; j <
data.output.health.health.health_services[0].mons.length; j++) {
                        mon2 =
data.output.health.health.health_services[0].mons[j];
                        if (mon.name == mon2.name) {
                            for (key in mon2) {
                                if (( key == "health") && (mon
[key]+"" != "undefined"))
                                    mon[key] =healthCompare(mon
[key],mon2[key]);
                                else
                                    mon[key] = mon2[key];
                            }
                            break
                        }
                    }
                }

                function healthCompare(h1,h2){
                    if ((h1 == "HEALTH_ERROR")||(h2 ==
"HEALTH_ERROR")) return "HEALTH_ERROR";
                    if ((h1 == "HEALTH_WARN")||(h2 ==
"HEALTH_WARN")) return "HEALTH_WARN";
                    if ((h1 == "HEALTH_OK")||(h2 == "HEALTH_OK"))
return "HEALTH_OK";
                    return "HEALTH_UNKNOWN";
                }

                var osdmap = data.output.osdmap.osdmap;
                $scope.osdsUp = osdmap.num_up_osds;
                $scope.osdsIn = parseInt(osdmap.num_in_osds);
                $scope.osdsOut = osdmap.num_osds -
osdmap.num_in_osds;
                $scope.osdsDown = $scope.osdsIn - $scope.osdsUp +
```

```
$scope.osdsOut;
                $scope.osdsNearFull = osdmap.nearfull?1:0;
                $scope.osdsFull = osdmap.full?1:0;
            })
            .error(function (data) {
                $scope.health = {};
                $scope.health.severity = "HEALTH_WARN";
                $scope.health.summary = "Status not available";
            });
    }

    $scope.refreshPGcontrol = function(){
        $scope.refreshPG = !$scope.refreshPG;
        if ($scope.refreshPG)
            $scope.refreshPGcontrolClass = "icon-pause";
        else
            $scope.refreshPGcontrolClass = "icon-play";
    }

    $scope.badgeClass = function (type, count) {
        if ((count == 0) || (count + "" == "undefined"))
            return "health_status_HEALTH_UNKNOWN mybadge";
        else
            return "health_status_HEALTH_" + type + " mybadge";
    }

    $scope.showModule = function(module,view){
        $cookieStore.put(module,view);
        $scope[module]=view;
    }

    $scope.getPgmapMessage=function(){
        if (typeof  $scope.pgmap ==="undefined") return "";
        var message = "";
        if ((typeof  $scope.pgmap.degraded_objects !=="undefined"
)&&($scope.pgmap.degraded_objects!=0)) {
            if (message != "") message += ", ";
            message += $scope.pgmap.degraded_objects + " objects
degraded on " + $scope.pgmap.degraded_total + " (" + (100 *
$scope.pgmap.degraded_objects /
$scope.pgmap.degraded_total).toFixed(3) + "%)";
        }
        if ((typeof  $scope.pgmap.misplaced_objects !
=="undefined" )&&($scope.pgmap.misplaced_objects!=0)) {
            if (message != "") message += ", ";
            message += $scope.pgmap.misplaced_objects +" objects
misplaced on "+$scope.pgmap.misplaced_total +" ("+ (100*
$scope.pgmap.misplaced_objects/
$scope.pgmap.misplaced_total).toFixed(3) +"%)";
        }
        return message;
    }
```

```
    function historise() {
        if ($scope.last_health_summary + "" == "undefined") {
            $scope.last_health_summary = $scope.health.summary;
            $scope.journal.unshift({"date": new Date(),
"summary": $scope.health.summary});
            return;
        }
        if ($scope.last_health_summary != $scope.health.summary)
{
            $scope.journal.unshift({"date": new Date(),
"summary": $scope.health.summary});
            $scope.last_health_summary = $scope.health.summary;
        }
    }

});

StatusApp.controller('ModalDemoCtrl', function ($scope, $modal,
$log) {
    $scope.open = function (size) {

        var modalInstance = $modal.open({
            templateUrl: 'myModalContent.html',
            controller: 'ModalInstanceCtrl',
            size: size
        });

        modalInstance.result.then(function (selectedItem) {
            $scope.selected = selectedItem;
        }, function () {
            $log.info('Modal dismissed at: ' + new Date());
        });
    };
});

// Please note that $modalInstance represents a modal window
(instance) dependency.
// It is not the same as the $modal service used above.

StatusApp.controller('ModalInstanceCtrl', function ($scope,
$modalInstance, $http) {
    $scope.refresh = function () {
        $scope.details = ["fetching details ..."];
        $http({method: "get", url: cephRestApiURL + "health.json?
detail",timeout:8000})
            .success(function (data, status) {
                $scope.details = data.output.detail;
                for (var i=0;i<$scope.details.length;i++){
                    var txt = $scope.details[i];
                    /(osd\.[0-9]+)/.exec(txt);
                    var osd = RegExp.$1;
```

```
                    if (osd != null) {
                        txt = txt.replace(osd, "<a
href='osds.html?dispoMode=space&" + osd + "'>" + osd + "</a>");
                    }
                   /([0-9]+\.[0-9a-f]+)/.exec(txt);
                    var id = RegExp.$1;
                    if (id != null){
                        //$scope.details[i] = txt.replace(id,"<a
href='pg.html?id="+id+"'>"+id+"</a>");
                        txt = txt.replace("pg "+id,"<a
href='../ceph-rest-api/tell/"+id+"/query.json'>pg "+id+"</a>");
                        }
                     // ceph-rest-api/tell/<id>/query.json
                    $scope.details[i] = txt;

                }
            })
            .error(function (data, status) {
                $scope.details = ["details not available"];
            }
        );
    };
    $scope.refresh();

    $scope.close = function () {
        $modalInstance.dismiss('cancel');
    };
});
```