const { config } = require("../config");

const { fetchJson, postJson } = require("../api");

var fs = require("file-system");

const \_ = require("lodash");

var fileName = "DashboardSharing\_" + Date.now() + ".txt";

const serverUrl = config.prodUrl;

var metadata = "";

var metadata2 = "";

var itemQuery = "";

var fileData = "";

var resource = "";

var type = "";

var resourceId = "";

var sharingPayload = "";

var userOwnerDashboard = null;

var diferencia2 = null;

var diferenciaUserGroups = null;

// IMPORTANT: Execute with a user that has the right to see the dashboard

(async () => {

    const  dashboardQuery  =  "dashboards.json?paging=false&fields=name,id,user[displayName,id],userAccesses,userGroupAccesses,dashboardItems&filter=id:in:[XJn1L2F87Rr]";

    metadata2 = await fetchJson({ apiQuery: dashboardQuery, url: serverUrl });

    for (index in metadata2.dashboards) {

        var dashboard = metadata2.dashboards[index];

        userOwnerDashboard = {

            "access": "rw------",

            "displayName": dashboard.user.displayName,

            "id": dashboard.user.id,

            "userUid": dashboard.user.id

        }

        dashboard.userAccesses.push(userOwnerDashboard);

        var userAccesses = \_.orderBy(dashboard.userAccesses, ["id"]);

        var userGroupAccesses = \_.orderBy(dashboard.userGroupAccesses, ["id"]);

        var dashboardItems = dashboard.dashboardItems;

        console.log(dashboard.name);

        for (index2 in dashboardItems) {

            var dashboardItem = dashboardItems[index2];

            metadata = "";

            resource = "";

            type = "";

            resourceId = "";

            itemQuery = "";

            sharingPayload = "";

            diferencia2 = null;

            /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

            /\*              REPORT TABLES                     \*/

            /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

            if (dashboardItem.reportTable != undefined) {

                type = "reportTable";

                resourceId = dashboardItem.reportTable.id;

                itemQuery = "metadata.json?reportTable=true&filter=id:eq:" + resourceId;

                metadata = await fetchJson({ apiQuery: itemQuery, url: serverUrl });

                resource = metadata.reportTables[0];

            }

            /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

            /\*              EVENT REPORTS                     \*/

            /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

            if (dashboardItem.eventReport != undefined) {

                type = "eventReport";

                resourceId = dashboardItem.eventReport.id;

                itemQuery = "metadata.json?eventReport=true&filter=id:eq:" + resourceId;

                metadata = await fetchJson({ apiQuery: itemQuery, url: serverUrl });

                resource = metadata.eventReports[0];

            }

            /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

            /\*              EVENT CHARTS                     \*/

            /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

            if (dashboardItem.eventChart != undefined) {

                type = "eventChart";

                resourceId = dashboardItem.eventChart.id;

                itemQuery = "metadata.json?eventChart=true&filter=id:eq:" + resourceId;

                metadata = await fetchJson({ apiQuery: itemQuery, url: serverUrl });

                resource = metadata.eventCharts[0];

            }

            /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

            /\*              CHARTS                            \*/

            /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

            if (dashboardItem.chart != undefined) {

                type = "chart";

                resourceId = dashboardItem.chart.id;

                itemQuery = "metadata.json?charts=true&filter=id:eq:" + resourceId;

                metadata = await fetchJson({ apiQuery: itemQuery, url: serverUrl });

                resource = metadata.charts[0];

            }

            if (resource != "") {

                resourceName = resource.name;

                resource.userAccesses.push(userOwnerDashboard);

                resourceGroupAccesses = \_.orderBy(resource.userGroupAccesses, ["id"]);

                mergedGroupAccesses = merge(userGroupAccesses, resourceGroupAccesses);

                userGroupAccesses = update(mergedGroupAccesses, userGroupAccesses);

                resourceuserAccesses = \_.orderBy(resource.userAccesses, ["id"]);

                merged = merge(userAccesses, resourceuserAccesses);

                userAccesses = update(merged, userAccesses);

                /\*              reportTable                            \*/

                if (dashboardItem.reportTable != undefined) {

                    metadata.reportTables[0].userAccesses = merged;

                    metadata.reportTables[0].userGroupAccesses = mergedGroupAccesses;

                }

                /\*              Event Report                    \*/

                if (dashboardItem.eventReport != undefined) {

                    metadata.eventReports[0].userAccesses = merged;

                    metadata.eventReports[0].userGroupAccesses = mergedGroupAccesses;

                }

                /\*              Event Chart                    \*/

                if (dashboardItem.eventChart != undefined) {

                    metadata.eventCharts[0].userAccesses = merged;

                    metadata.eventCharts[0].userGroupAccesses = mergedGroupAccesses;

                }

                /\*              chart                                \*/

                if (dashboardItem.chart != undefined) {

                    metadata.charts[0].userAccesses = merged;

                    metadata.charts[0].userGroupAccesses = mergedGroupAccesses;

                }

                sharingPayload =

                    {

                        "object": {

                            "id": resourceId,

                            "publicAccess": "--------",

                            "externalAccess": false,

                            "userGroupAccesses": mergedGroupAccesses,

                            "userAccesses": merged

                        }

                    }

                diferencia2 = \_.differenceWith(userAccesses, resourceuserAccesses, \_.isEqual);

                diferenciaUserGroups = \_.differenceWith(userGroupAccesses, resourceGroupAccesses, \_.isEqual);

                fileData = logAcceses(diferencia2, fileName, dashboard, "USER", type, resource, resourceName, resourceId);

                fileDataUserGroups = logAcceses(diferenciaUserGroups, fileName, dashboard, "USER GROUP", type, resource, resourceName, resourceId);

                if (fileData == undefined) { fileData = ""; }

                if (fileDataUserGroups != undefined) {

                    fileData += fileDataUserGroups;

                }

                postQuery = "sharing?type=" + type + "&id=" + resourceId;

                if (diferencia2.length > 0 || diferenciaUserGroups.length > 0) {

                    response = await postJson({ apiQuery: postQuery, body: sharingPayload, url: serverUrl });

                    if (response != undefined) {

                        fileData += (JSON.stringify(response));

                    }

                }

                if (fileData != undefined && (diferencia2.length > 0 || diferenciaUserGroups.length > 0)) {

                    console.log(fileData);

                }

            }

        }

    }

    function update(merged, userAccesses) {

        for (i in userAccesses) {

            for (i2 in merged) {

                if (userAccesses[i].id == merged[i2].id) {

                    userAccesses[i].access = merged[i2].access

                }

            }

        }

        return userAccesses;

    }

    function merge(dashboardAccesses, resourceAccesses) {

        var m = \_.unionBy(dashboardAccesses, resourceAccesses, "id");

        for (var i in m) {

            id = m[i].id;

            access = m[i].access;

            for (i2 in resourceAccesses) {

                if (resourceAccesses[i2].id == id && resourceAccesses[i2].access == "rw------" && access == "r-------") {

                      m[i].access = "rw------";

                }

            }

        }

        return m;

    }

    function logAcceses(diferencia2, filename, dashboard, userType, type, resource, chartName, chartId) {

        var fileData = "";

        if (diferencia2.length > 0) {

            fileData = "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

            fileData += "\r\n";

            fileData += "Dashboard :" + dashboard.name + " " + dashboard.id;

            fileData += "\r\n";

            fileData += type + " " + chartName + " " + chartId;

            fileData += "\r\n";

            fileData += "\_\_\_\_\_\_\_\_\_\_\_\_\_\_" + userType + " DIFFERENCES\_\_\_\_\_\_\_\_\_\_\_\_\_\_";

            fileData += "\r\n";

            for (index3 in diferencia2) {

                fileData +=

                    userType + " Id: " + diferencia2[index3].id + " " +

                    userType + ": " + diferencia2[index3].displayName +

                    " Access: " + diferencia2[index3].access;

                fileData += "\r\n";

            }

            fileData += "\r\n";

            fileData += "New " + type + " " + chartName + " " + chartId;

            fileData += "\r\n";

            fileData += "\_\_\_\_\_\_\_\_\_\_NEW " + userType;

            fileData += "\r\n";

            if (userType == "USER GROUP") {

                for (index3 in resource.userGroupAccesses) {

                    fileData +=

                        userType + " Id: " + resource.userGroupAccesses[index3].id + " " +

                        userType + ": " + resource.userGroupAccesses[index3].displayName +

                        " Access: " + resource.userGroupAccesses[index3].access;

                    fileData += "\r\n";

                }

            }

            if (userType == "USER") {

                for (index3 in resource.userAccesses) {

                    fileData +=

                        userType + " Id: " + resource.userAccesses[index3].id + " " +

                        userType + ": " + resource.userAccesses[index3].displayName +

                        " Access: " + resource.userAccesses[index3].access;

                    fileData += "\r\n";

                }

            }

            fileData += "\r\n";

            fs.appendFile(fileName, fileData, err => {

                if (err) console.log(err);

            });

            fileData += "Successfully Written to File.";

            return fileData;

        }

    }

})();