import java.io.IOException;

import javax.jcr.Node;

import javax.jcr.NodeIterator;

import javax.jcr.RepositoryException;

import javax.jcr.Session;

import javax.jcr.query.Query;

import javax.jcr.query.QueryManager;

import javax.jcr.query.QueryResult;

import javax.servlet.ServletException;

import org.apache.felix.scr.annotations.Reference;

import org.apache.felix.scr.annotations.sling.SlingServlet;

import org.apache.sling.api.SlingHttpServletRequest;

import org.apache.sling.api.SlingHttpServletResponse;

import org.apache.sling.api.resource.ResourceResolver;

import org.apache.sling.api.resource.ResourceResolverFactory;

import org.apache.sling.api.servlets.SlingAllMethodsServlet;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

@SlingServlet(paths = "/bin/Propertydelete", methods = "GET")

public class PropertyDeleteServlet extends SlingAllMethodsServlet {

/\*\*

 \*

 \*/

private static final long serialVersionUID = 1L;

private static final Logger LOGGER = LoggerFactory.getLogger(PropertyDeleteServlet.class);

public String path = "/content/marsh";

@Reference

public ResourceResolverFactory resourceResolverFactory;

@Override

protected void doGet(final SlingHttpServletRequest request, final SlingHttpServletResponse response)

throws ServletException, IOException {

try {

@SuppressWarnings("deprecation")

ResourceResolver resourceResolver = resourceResolverFactory.getAdministrativeResourceResolver(null);

Session session = resourceResolver.adaptTo(Session.class);

String copyRightDateQuery = "SELECT \* FROM [nt:base] AS s WHERE ISDESCENDANTNODE([/content]) and [copyrightDate] is not null";

QueryManager queryManager;

queryManager = session.getWorkspace().getQueryManager();

Query query = queryManager.createQuery(copyRightDateQuery, Query.JCR\_SQL2);

QueryResult copyRightDate = query.execute();

NodeIterator assetIterator = copyRightDate.getNodes();

while (assetIterator.hasNext()) {

Node copyRightDateNode = (Node) assetIterator.nextNode();

if (!(copyRightDateNode.getPath().contains(path))) {

copyRightDateNode.getProperty("copyrightDate").remove();

response.getWriter().print("deleting the property copyrightDate from the node :  ");

response.getWriter().println(copyRightDateNode.getPath());

session.save();

}

}

} catch (RepositoryException e) {

LOGGER.error("error in getting session", e);

} catch (Exception e) {

LOGGER.error("error in getting session", e);

}

}

}

On Wed, Jun 7, 2017 at 10:28 PM, Chetan Kumar <[bck2301@gmail.com](mailto:bck2301@gmail.com)> wrote:

package com.regnant.ServletSample;  
  
import java.io.IOException;  
  
import javax.jcr.Node;  
import javax.jcr.NodeIterator;  
import javax.jcr.RepositoryException;  
import javax.jcr.Session;  
import javax.jcr.query.Query;  
import javax.jcr.query.QueryManager;  
import javax.jcr.query.QueryResult;  
import javax.servlet.ServletException;  
  
import org.apache.felix.scr.annotations.Reference;  
import org.apache.felix.scr.annotations.sling.SlingServlet;  
import org.apache.sling.api.SlingHttpServletRequest;  
import org.apache.sling.api.SlingHttpServletResponse;  
import org.apache.sling.api.servlets.SlingAllMethodsServlet;  
import org.apache.sling.jcr.api.SlingRepository;  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
  
import com.day.cq.replication.ReplicationActionType;  
import com.day.cq.replication.ReplicationException;  
import com.day.cq.replication.Replicator;  
import com.day.cq.workflow.WorkflowException;  
import com.day.cq.workflow.WorkflowService;  
import com.day.cq.workflow.WorkflowSession;  
import com.day.cq.workflow.exec.WorkflowData;  
import com.day.cq.workflow.model.WorkflowModel;  
  
  
@SlingServlet(paths = "/bin/property", methods = "GET")  
public class PropertyDeleteServlet extends SlingAllMethodsServlet {  
    /\*\*  
     \*   
     \*/  
    private static final long serialVersionUID = 1L;  
    private static final Logger LOGGER = LoggerFactory.getLogger(PropertyDeleteServlet.class);  
    public String path = "/content/customer";  
    @Reference  
    SlingRepository repo;  
      
  
    @Override  
    protected void doGet(final SlingHttpServletRequest request, final SlingHttpServletResponse response)  
            throws ServletException, IOException {  
        try {  
            @SuppressWarnings("deprecation")  
            Session session = repo.loginAdministrative(null);  
    String assetQuery = "SELECT \* FROM [nt:base] AS s WHERE ISDESCENDANTNODE([/content]) and [ABC] is not null";  
    QueryManager queryManager;  
    queryManager = session.getWorkspace().getQueryManager();  
    Query query = queryManager.createQuery(assetQuery, Query.JCR\_SQL2);  
    QueryResult assets = query.execute();  
    NodeIterator assetIterator = assets.getNodes();  
    while (assetIterator.hasNext()){  
        Node assetNode = (Node) assetIterator.nextNode();  
        response.getWriter().write("\*\*\*\*\*\*\*\*\*\*"+assetNode.getPath()+"\*\*\*\*\*\*\*\*\*\*\*\*\*");  
        if(!(assetNode.getPath().contains(path))){  
            assetNode.getProperty("ABC").remove();  
            session.save();  
        }  
    }  
    }catch(RepositoryException e){  
        LOGGER.error("error in getting session", e);  
    }catch(Exception e){  
        LOGGER.error("error in getting session", e);      
    }   
    }  
              
}