AEM best practices and development guide

Guidelines to improve code quality and avoid regressions

☰ Menu

Bulk Add, Update and Delete properties in AEM – without using Groovy console

Kiran Sg AEM 10th Dec 202129th Oct 2022 5 Minutes

Problem Statement:

How to Bulk Add, Update or remove page properties in AEM? Without using the Groovy console.

Requirement:

Create a reusable process that can be used to search for the pages based on resourceType and do the CRUD operations on the results.

Introduction:

Usually, whenever we are using editable templates, we might have some initial content but for some reason, if we want to update the experience fragment path or some page properties then usually, we go for Groovy script to run bulk update.

But AMS don't install developer tools on the PROD, we need to go to other options and for the above requirement, we can use MCP.

MCP (Manage Controlled Processes) is both a dashboard for performing complex tasks and a rich API for defining these tasks as process definitions. In addition to kicking off new processes, users can also monitor running tasks, retrieve information about completed tasks, halt work, and so on.

Add the following maven dependency to your pom to extend MCP

```
 <dependency>
    <groupId>com.adobe.acs</groupId>
    <artifactId>acs-aem-commons-bundle</artifactId>
    <version>5.0.4</version>
    <scope>provided</scope>
</dependency>
```

Create Process Definition factory – PropertyUpdateFactory

This class tells ACS Commons MCP to pick the process definition and process name **getName** and you need to mention the implementation class inside the **createProcessDefinitionInstance** method as shown below:

```java
package com.mysite.mcp.process;

import org.osgi.service.component.annotations.Component;
import com.adobe.acs.commons.mcp.ProcessDefinitionFactory;

@Component(service = ProcessDefinitionFactory.class, immediate = true)
public class PropertyUpdateFactory extends ProcessDefinitionFactory<PropertyUpdater> {
    @Override
    public String getName() {
        return "Property Updator";
    }
    @Override
    protected PropertyUpdater createProcessDefinitionInstance() {
        return new PropertyUpdater();
    }
}
```

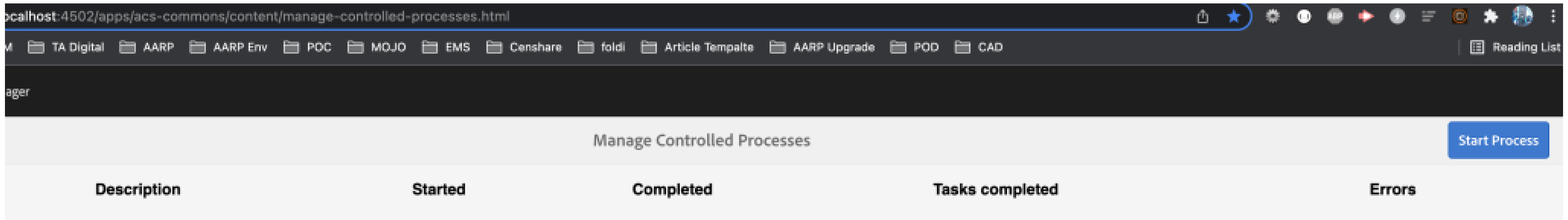
Create Process Definition implementation – PropertyUpdater

This is an implementation class where we are defining all the form fields required for the process to run

```java
package com.mysite.mcp.process;

import java.io.IOException;
import java.text.MessageFormat;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Collections;
import java.util.EnumMap;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import java.util.Objects;
import java.util.Optional;
import javax.jcr.RepositoryException;
import org.apache.commons.lang3.StringUtils;
import org.apache.sling.api.request.RequestParameter;
import org.apache.sling.api.resource.LoginException;
import org.apache.sling.api.resource.ModifiableValueMap;
import org.apache.sling.api.resource.PersistenceException;
import org.apache.sling.api.resource.Resource;
import org.apache.sling.api.resource.ResourceResolver;
import org.apache.sling.api.resource.ResourceUtil;
import org.apache.sling.resource.filter.ResourceFilterStream;
import org.jetbrains.annotations.NotNull;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import com.adobe.acs.commons.data.Spreadsheet;
import com.adobe.acs.commons.fam.ActionManager;
import com.adobe.acs.commons.mcp.ProcessDefinition;
import com.adobe.acs.commons.mcp.ProcessInstance;
import com.adobe.acs.commons.mcp.form.FileUploadComponent;
import com.adobe.acs.commons.mcp.form.FormField;
import com.adobe.acs.commons.mcp.form.SelectComponent;
import com.adobe.acs.commons.mcp.form.TextfieldComponent;
import com.adobe.acs.commons.mcp.model.GenericReport;
import com.day.cq.commons.jcr.JcrConstants;

public class PropertyUpdater extends ProcessDefinition {

    private static final Logger LOGGER = LoggerFactory.getLogger(PropertyUpdater.class);
    private final GenericReport report = new GenericReport();
    private static final String REPORT_NAME = "Property-update-report";
    private static final String RUNNING = "Running ";
    private static final String EXECUTING_KEYWORD = " Property Updation";

    private static final String PROPERTY = "property";
    private static final String PROPERTY_TYPE = "type";
    private static final String PROPERTY_VALUE = "value";

    protected enum UpdateAction {
        ADD, UPDATE, DELETE
    }

    @FormField(name = "Property Update Excel", component = FileUploadComponent.class)
    private RequestParameter sourceFile;

    @FormField(name = "Path",
            component = TextfieldComponent.class,
            hint = "(Provide the Relative path to start search)",
            description = "A query will be executed starting this path")
    private String path;

    @FormField(name = "Action",
            component = SelectComponent.EnumerationSelector.class,
            description = "Add, Update or Delete?",
            options = "default=Add")
    UpdateAction reAction = UpdateAction.ADD;

    @FormField(name = "ResourceType",
            component = TextfieldComponent.class,
            hint = "(Provide the resourcetype to be searched for)",
            description = "A query will be executed based on resourcetype")
    private String pageResourceType;

    private Map<String, Object> propertyMap = new HashMap<>();

    @Override
    public void init() throws RepositoryException {
        validateInputs();
    }

    @Override
    public void buildProcess(ProcessInstance instance, ResourceResolver rr) throws LoginException, RepositoryException {
        report.setName(REPORT_NAME);
        instance.getInfo().setDescription(RUNNING + reAction + EXECUTING_KEYWORD);
        instance.defineCriticalAction("Adding the Properties", rr, this::updateProperties);
    }

    private void updateProperties(ActionManager manager) {
        manager.deferredWithResolver(this::addProperties);
    }

    private void addProperties(ResourceResolver resourceResolver) {
        @NotNull Resource resource = resourceResolver.resolve(path);
        if(!ResourceUtil.isNonExistingResource(resource)) {
            ResourceFilterStream rfs = resource.adaptTo(ResourceFilterStream.class);
            rfs.setBranchSelector("[jcr:primaryType] == 'cq:Page'")
                    .setChildSelector("[jcr:content/sling:resourceType] == $type")
                    .addParam("type", pageResourceType)
                    .stream()
                    .map(r -> r.getChild(JcrConstants.JCR_CONTENT))
                    .forEach(this::updateProperty);
        }
    }

    private void updateProperty(Resource resultResource) {
        if (reAction == UpdateAction.ADD || reAction == UpdateAction.UPDATE) {
            addOrUpdateProp(resultResource);
        } else if (reAction == UpdateAction.DELETE) {
            removeProp(resultResource);
        }
    }

    private void removeProp(Resource resultResource) {
        try {
            ModifiableValueMap map = resultResource.adaptTo(ModifiableValueMap.class);
            propertyMap.entrySet().stream().forEach(r -> map.remove(r.getKey()));
            resultResource.getResourceResolver().commit();
            recordAction(resultResource.getPath(), reAction.name(), StringUtils.join(propertyMap));
        } catch (PersistenceException e) {
            LOGGER.error("Error occurred while persisting the property {}", e.getMessage());
        }
    }

    private void addOrUpdateProp(Resource resultResource) {
        try {
            ModifiableValueMap map = resultResource.adaptTo(ModifiableValueMap.class);
            propertyMap.entrySet().stream().forEach(r -> map.put(r.getKey(), r.getValue()));
            resultResource.getResourceResolver().commit();
            recordAction(resultResource.getPath(), reAction.name(), StringUtils.join(propertyMap));
        } catch (PersistenceException e) {
            LOGGER.error("Error occurred while persisting the property {}", e.getMessage());
        }
    }

    public enum ReportColumns {
        PATH, ACTION, DESCRIPTION
    }

    private void validateInputs() throws RepositoryException {
        if (sourceFile != null && sourceFile.getSize() > 0) {
            Spreadsheet sheet;
            try {
                sheet = new Spreadsheet(sourceFile, PROPERTY, PROPERTY_TYPE, PROPERTY_VALUE).buildSpreadsheet();
            } catch (IOException ex) {
                throw new RepositoryException("Unable to parse spreadsheet", ex);
            }

            if (!sheet.getHeaderRow().contains(PROPERTY) || !sheet.getHeaderRow().contains(PROPERTY_TYPE) || !sheet.getHeaderRow().contains(PROPERTY_V
                throw new RepositoryException(MessageFormat.format("Spreadsheet should have two columns, respectively named {0}, {1} and {2}", PROPERT
            }

            sheet.getDataRowsAsCompositeVariants().forEach(row -> {
                String propertyType = row.get(PROPERTY_TYPE).toString();
                if(StringUtils.equalsAnyIgnoreCase("String", propertyType)) {
                    propertyMap.put(row.get(PROPERTY).toString(), row.get(PROPERTY_VALUE).toString());
                } else if(StringUtils.equalsAnyIgnoreCase("Date", propertyType)) {
                    SimpleDateFormat dt = new SimpleDateFormat("yyyy-MM-dd'T'HH:mm:ss.SSS");
                    propertyMap.put(row.get(PROPERTY).toString(), dt.format(row.get(PROPERTY_VALUE).toString()));
                } else if(StringUtils.equalsAnyIgnoreCase("Array", propertyType)) {
                    propertyMap.put(row.get(PROPERTY).toString(), row.get(PROPERTY_VALUE).toString().split(","));
                } else if(StringUtils.equalsAnyIgnoreCase("Long", propertyType)) {
                    Integer result = Optional.ofNullable(row.get(PROPERTY_VALUE).toString())
                            .filter(Objects::nonNull)
                            .map(Integer::parseInt)
                            .orElse(0);
                    propertyMap.put(row.get(PROPERTY).toString(), result);
                } else if(StringUtils.equalsAnyIgnoreCase("Binary", propertyType)) {
                    propertyMap.put(row.get(PROPERTY).toString(), Boolean.valueOf(row.get(PROPERTY_VALUE).toString()));
                }
            });
        }
    }

    List<EnumMap<ReportColumns, String>> reportData = Collections.synchronizedList(new ArrayList<>());

    private void recordAction(String path, String action, String description) {
        EnumMap<ReportColumns, String> row = new EnumMap<>(ReportColumns.class);
        row.put(ReportColumns.PATH, path);
        row.put(ReportColumns.ACTION, action);
        row.put(ReportColumns.DESCRIPTION, description);
        reportData.add(row);
    }

    @Override
    public void storeReport(ProcessInstance instance, ResourceResolver rr) throws RepositoryException, PersistenceException {
        report.setRows(reportData, ReportColumns.class);
        report.persist(rr, instance.getPath() + "/jcr:content/report");
    }
}
```

Once code is deployed, please go to the following URL and click on start process as shown below:

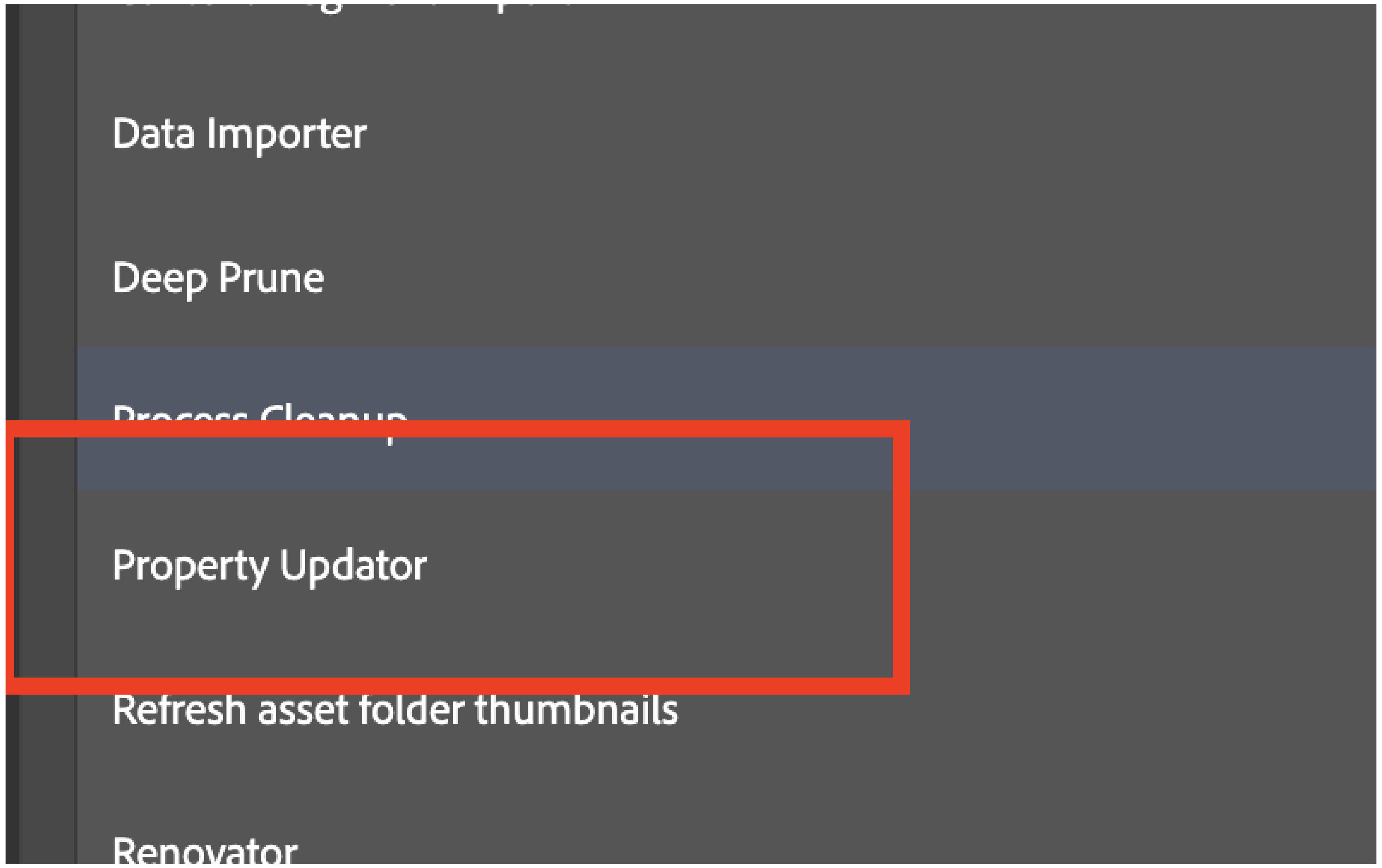
http:// (http://){domain}/apps/acs-commons/content/manage-controlled-processes.html



Start MCP Process

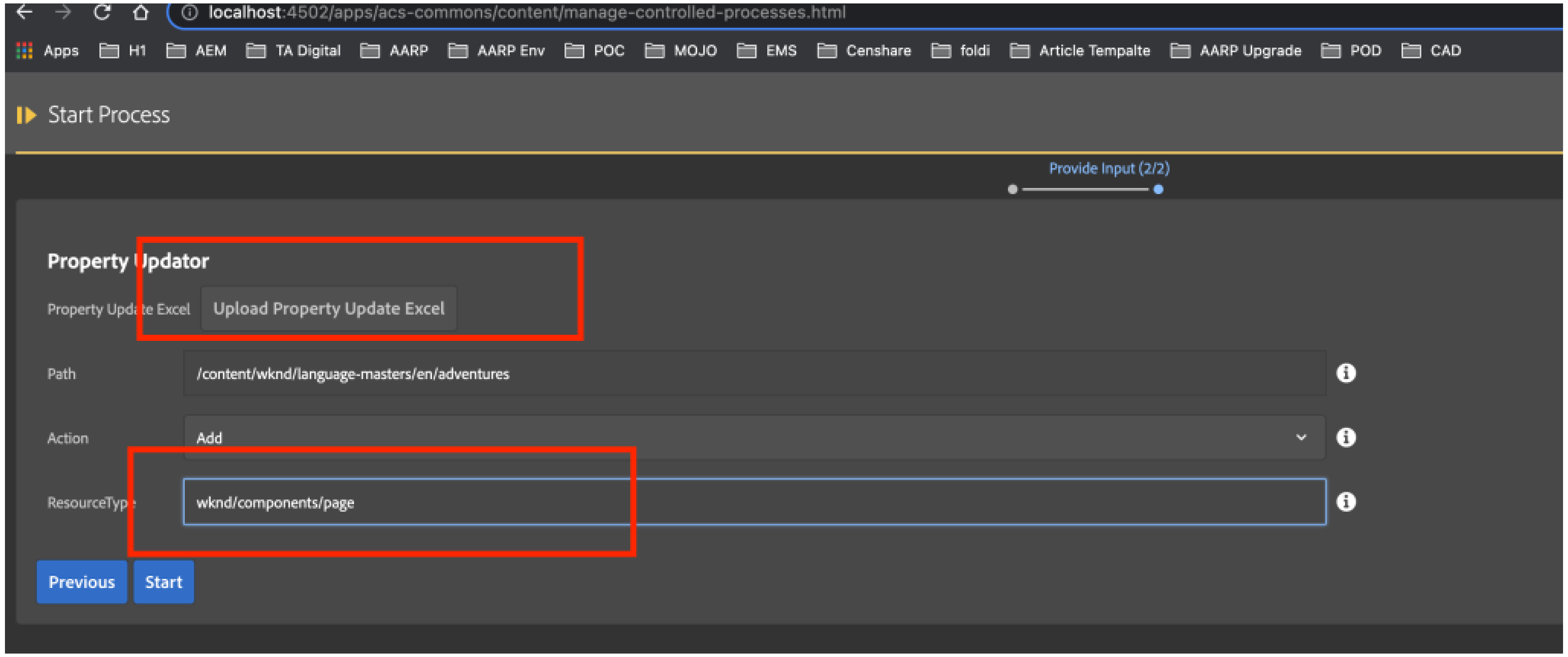
You will see a new process called Property Update as shown below:

Click on the process



Property Updater process

Upload the excel sheet and provide search Path, Action and Resource Type to search for



localhost:4502/apps/acs-commons/content/manage-controlled-processes.html

Property Updator

Property Update Excel: Upload Property Update Excel

Path: /content/wknd/language-masters/en/adventures

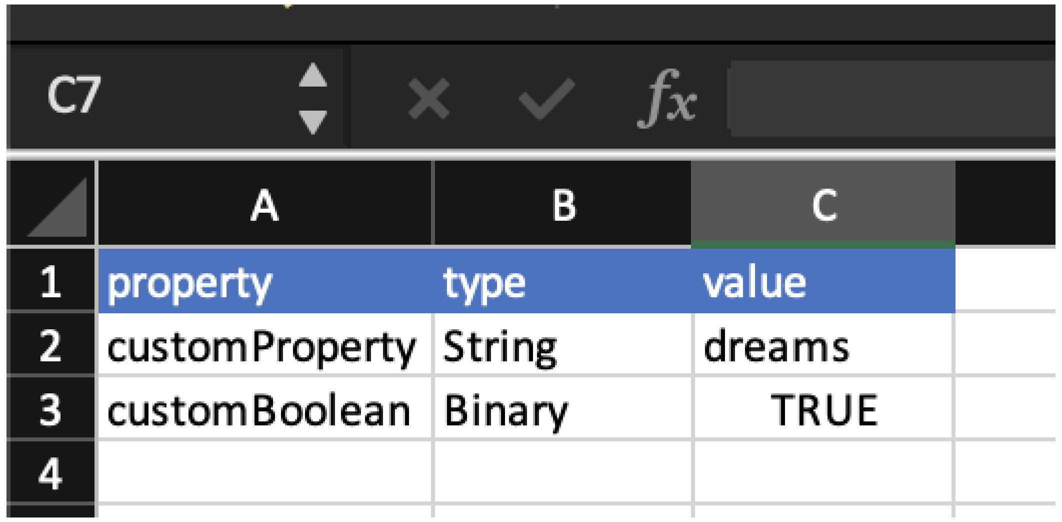
Action: Add

ResourceType: wknd/components/page

Configure Process

Excel Sheet:

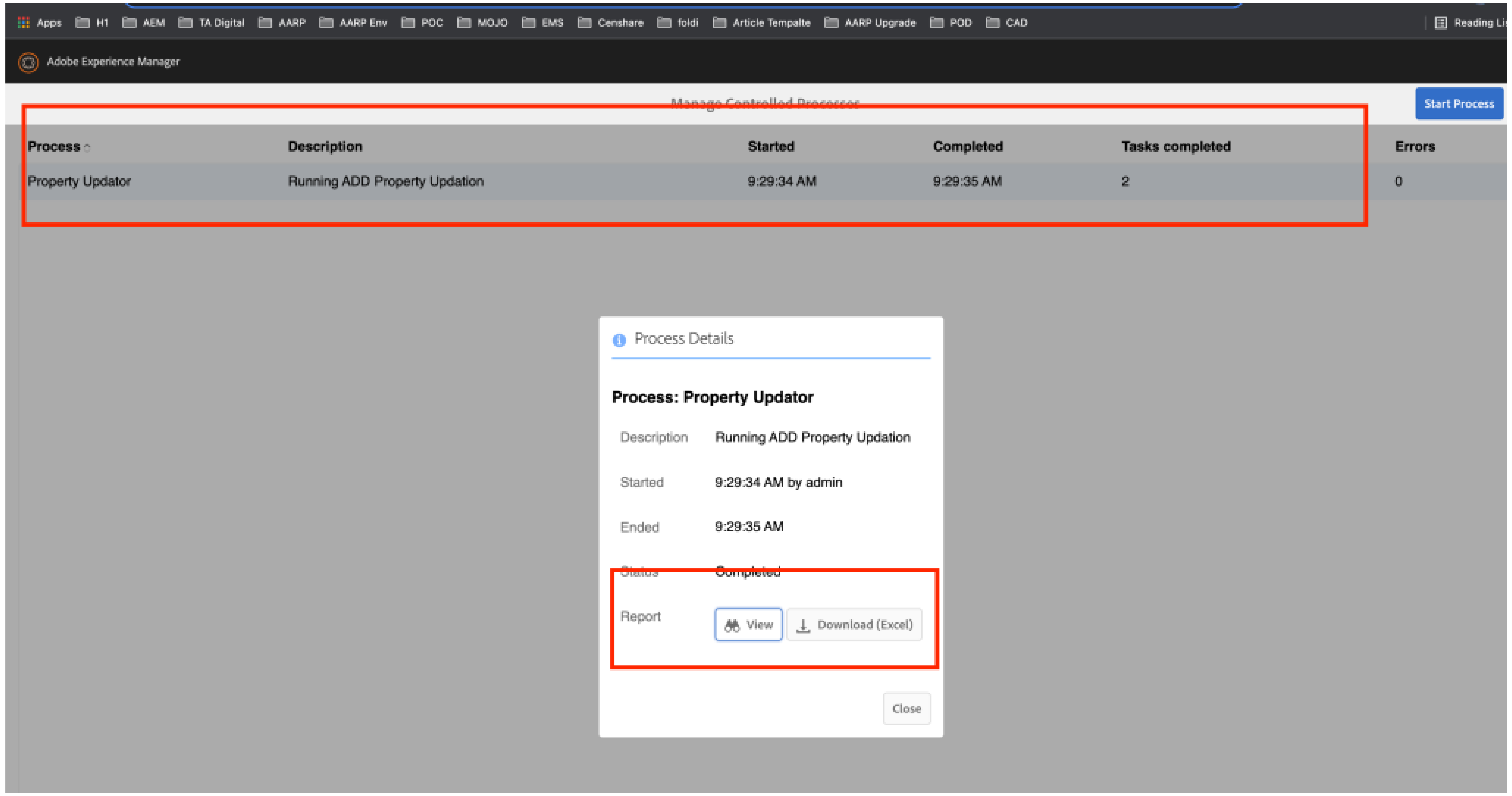
Upload the excel sheet attached and add the property name, data type and value as shown below:



	A	B	C
1	property	type	value
2	customProperty	String	dreams
3	customBoolean	Binary	TRUE
4			

Property Updater Excel

propertyupdater (https://kirantech58867409.files.wordpress.com/2021/12/propertyupdater.xlsx) Download (https://kirantech58867409.files.wordpress.com/2021/12/propertyupdater.xlsx?force_download=true)
After executing the process, you can click the completed process and view the results as shown below:



Process	Description	Started	Completed	Tasks completed	Errors
Property Updator	Running ADD Property Updation	9:29:34 AM	9:29:35 AM	2	0

Process Details

Process: Property Updator

Description Running ADD Property Updation

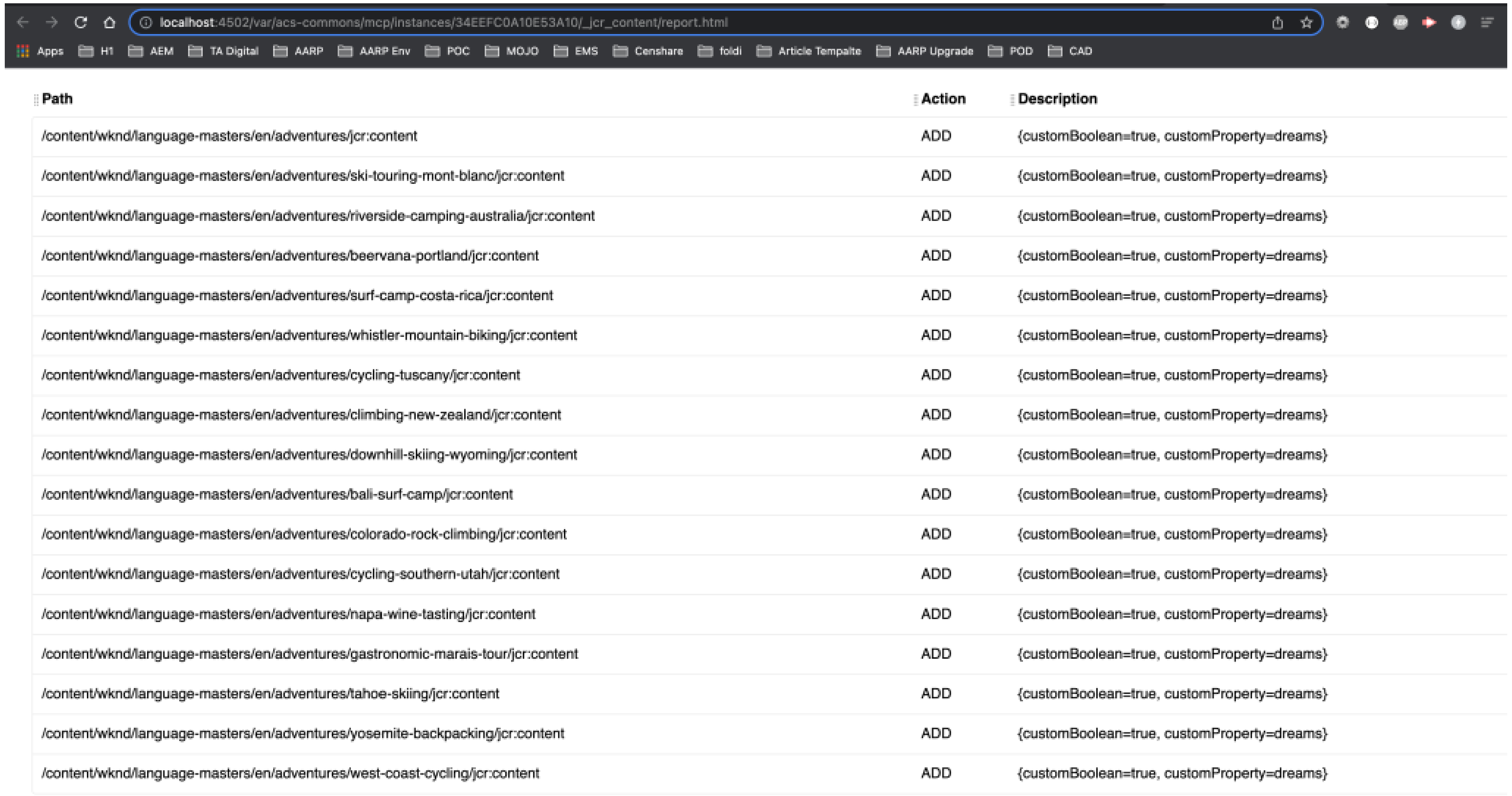
Started 9:29:34 AM by admin

Ended 9:29:35 AM

Status Completed

Report

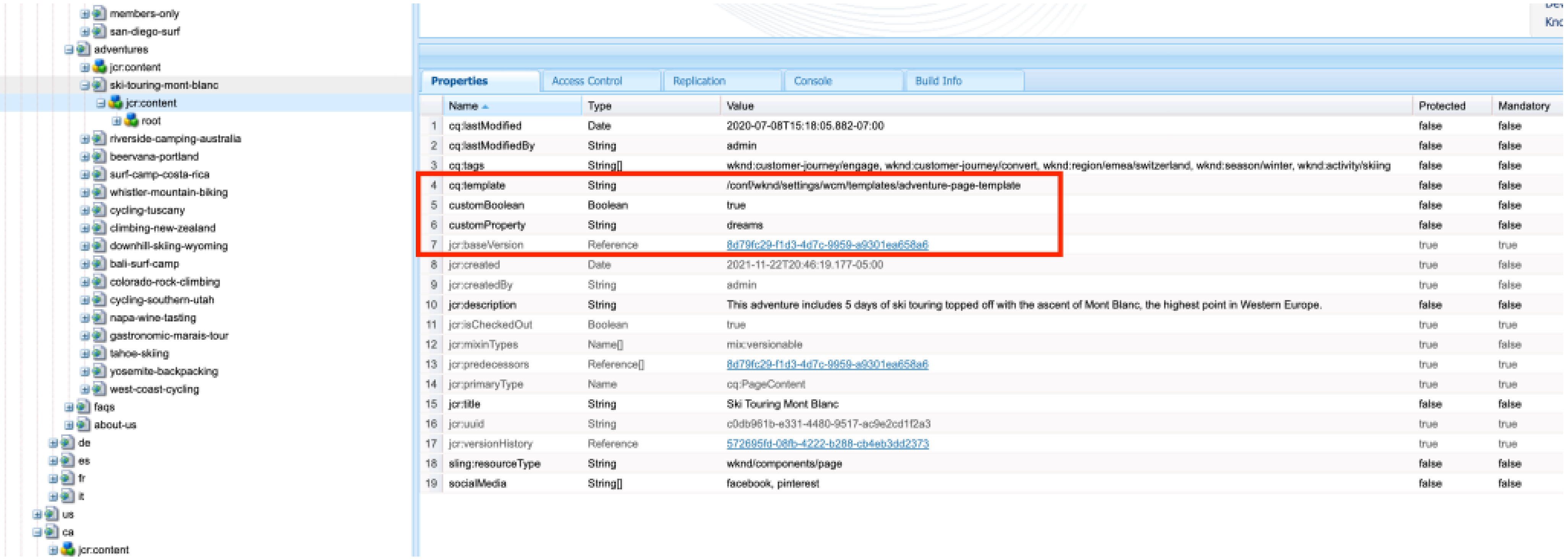
View report



Path	Action	Description
/content/wknd/language-masters/en/adventures/jcr:content	ADD	{customBoolean=true, customProperty=dreams}
/content/wknd/language-masters/en/adventures/ski-touring-mont-blanc/jcr:content	ADD	{customBoolean=true, customProperty=dreams}
/content/wknd/language-masters/en/adventures/riverside-camping-australia/jcr:content	ADD	{customBoolean=true, customProperty=dreams}
/content/wknd/language-masters/en/adventures/beervana-portland/jcr:content	ADD	{customBoolean=true, customProperty=dreams}
/content/wknd/language-masters/en/adventures/surf-camp-costa-rica/jcr:content	ADD	{customBoolean=true, customProperty=dreams}
/content/wknd/language-masters/en/adventures/whistler-mountain-biking/jcr:content	ADD	{customBoolean=true, customProperty=dreams}
/content/wknd/language-masters/en/adventures/cycling-tuscany/jcr:content	ADD	{customBoolean=true, customProperty=dreams}
/content/wknd/language-masters/en/adventures/climbing-new-zealand/jcr:content	ADD	{customBoolean=true, customProperty=dreams}
/content/wknd/language-masters/en/adventures/downhill-skiing-wyoming/jcr:content	ADD	{customBoolean=true, customProperty=dreams}
/content/wknd/language-masters/en/adventures/bali-surf-camp/jcr:content	ADD	{customBoolean=true, customProperty=dreams}
/content/wknd/language-masters/en/adventures/colorado-rock-climbing/jcr:content	ADD	{customBoolean=true, customProperty=dreams}
/content/wknd/language-masters/en/adventures/cycling-southern-utah/jcr:content	ADD	{customBoolean=true, customProperty=dreams}
/content/wknd/language-masters/en/adventures/napa-wine-tasting/jcr:content	ADD	{customBoolean=true, customProperty=dreams}
/content/wknd/language-masters/en/adventures/gastronomic-marais-tour/jcr:content	ADD	{customBoolean=true, customProperty=dreams}
/content/wknd/language-masters/en/adventures/tahoe-skiing/jcr:content	ADD	{customBoolean=true, customProperty=dreams}
/content/wknd/language-masters/en/adventures/yosemite-backpacking/jcr:content	ADD	{customBoolean=true, customProperty=dreams}
/content/wknd/language-masters/en/adventures/west-coast-cycling/jcr:content	ADD	{customBoolean=true, customProperty=dreams}

List of updated results with Action

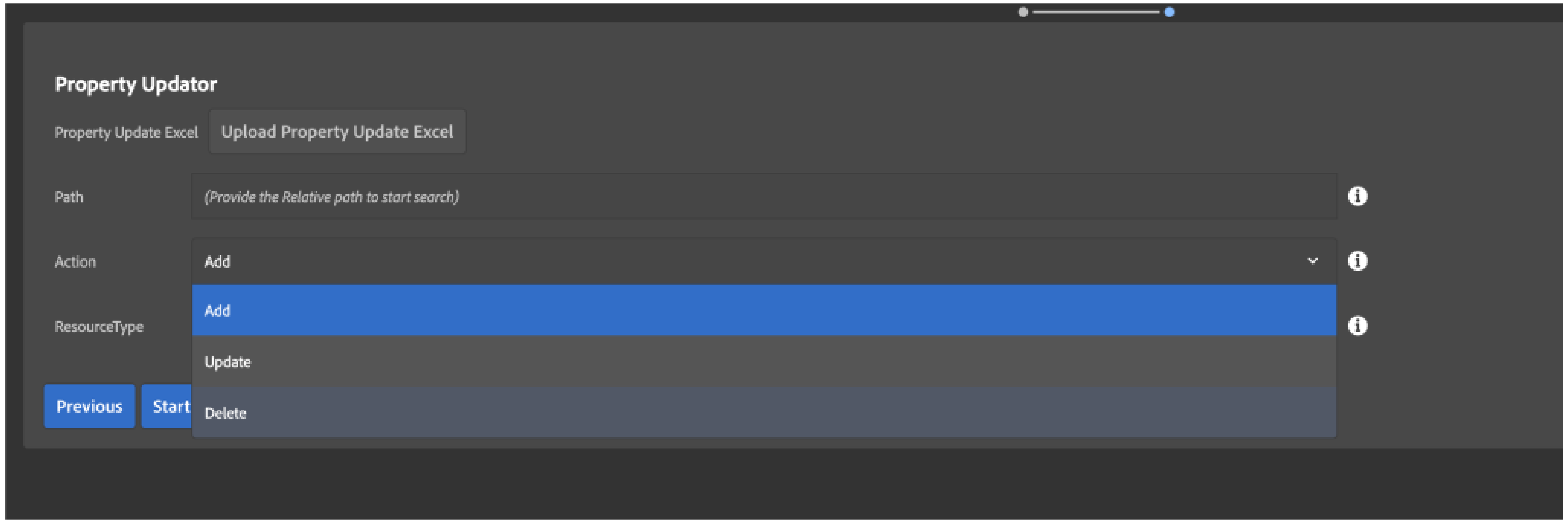
You can also go to CRXDE and check the property getting added to the resources as shown below:



	Name	Type	Value	Protected	Mandatory
1	cq:lastModified	Date	2020-07-08T15:18:05.882-07:00	false	false
2	cq:lastModifiedBy	String	admin	false	false
3	cq:tags	String[]	wknd:customer-journey/engage, wknd:customer-journey/convert, wknd:region/emea/switzerland, wknd:season/winter, wknd:activity/skiing	false	false
4	cq:template	String	/conf/wknd/settings/wcm/templates/adventure-page-template	false	false
5	customBoolean	Boolean	true	false	false
6	customProperty	String	dreams	false	false
7	jcr:baseVersion	Reference	8d79fc29-f1d3-4d7c-9959-a9301ea658a6	true	true
8	jcr:created	Date	2021-11-22T20:46:19.177-05:00	true	false
9	jcr:createdBy	String	admin	true	false
10	jcr:description	String	This adventure includes 5 days of ski touring topped off with the ascent of Mont Blanc, the highest point in Western Europe.	false	false
11	jcr:isCheckedOut	Boolean	true	true	true
12	jcr:mixinTypes	Name[]	mix:versionable	true	false
13	jcr:predecessors	Reference[]	8d79fc29-f1d3-4d7c-9959-a9301ea658a6	true	true
14	jcr:primaryType	Name	cq:PageContent	true	true
15	jcr:title	String	Ski Touring Mont Blanc	false	false
16	jcr:uuid	String	c0db961b-e331-4480-9517-ac9e2cd1f2a3	true	true
17	jcr:versionHistory	Reference	572695fd-08fb-4222-b288-cb4eb3dd2373	true	true
18	sling:resourceType	String	wknd/components/page	false	false
19	socialMedia	String[]	facebook, pinterest	false	false

CRXDE results

You can also perform other options by selecting appropriate Action as shown below:



Property Updator

Property Update Excel: Upload Property Update Excel

Path: (Provide the Relative path to start search)

Action: Add

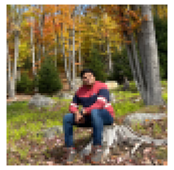
- Add
- Update
- Delete

ResourceType

Action Dropdown

You can learn more about MCP here (https://adobe-consulting-services.github.io/acs-aem-commons/features/mcp/index.html)

Tagged:
ACS Commons,
AEM,
AEM as a Cloud Service,
aemaacs,
Groovy,
Java,
MCP,
Property



Published by Kiran Sg

My name is Kiran SG. I am currently working as an AEM architect and handling site revamp and migration. I try to learn new tricks and techniques every day to improve my coding and deliver value-added projects. All the tricks and techniques sometimes seem to be straightforward working code with best practices. But sometimes it won't satisfy the developer's needs. But most of the time code is either lagging to meet the requirement or doesn't follow best practices. Hence I am taking the initiative to put all my learnings and tricks in my blog series to share the working code with best practices. Hoping to help other developers like me. I have been working on AEM for 8 years now. I started my career as a Java developer. I have delivered many projects with a core customer-centric and strong focus on infrastructure and architecture. If you have any queries related to my blog topics please do connect with me directly on my email and also you can connect with me over LinkedIn and Facebook. *View all posts by Kiran Sg*

3 thoughts on “Bulk Add, Update and Delete properties in AEM – without using Groovy console”

Kunal says:
28th Mar 2023 at 4:33 am
import org.apache.sling.resource.filter.ResourceFilterStream;
I am getting error for the above import. Could you please help how to get that fixed.

Reply

> **Kiran Sg** says:
> 30th Mar 2023 at 10:07 am
> Add dependencies
>
> Reply

Rahul says:
10th Apr 2023 at 9:24 am
Is there a junit available for this or are you planning to create?

Reply