

Task-12 CODE

Question:

1. When you create any page in the project, you should trigger the event listener. After that, you should trigger the workflow, and it should add the Expirydate property to that page's jcr:content, with the value being the current date and time.

2.Continuation: And when you create any page in another project, it should add the Expirydate property to that page's jcr:content, with the value being the previous date and time.

Answer :

EVENT LISTENER FOR CREATING PAGE WORKFLOW STARTED

```
package com.adobe.aem.guides.demo.core.listeners;

import java.util.HashMap;
import java.util.Map;

import javax.jcr.Session;
import javax.jcr.observation.Event;
import javax.jcr.observation.EventIterator;
import javax.jcr.observation.EventListener;

import org.apache.sling.api.resource.LoginException;
import org.apache.sling.api.resource.Resource;
import org.apache.sling.api.resource.ResourceResolver;
import org.apache.sling.api.resource.ResourceResolverFactory;
import org.apache.sling.jcr.api.SlingRepository;
import org.osgi.service.component.annotations.Activate;
import org.osgi.service.component.annotations.Component;
import org.osgi.service.component.annotations.Deactivate;
import org.osgi.service.component.annotations.Reference;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
```

```
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;
```

```
@Component(immediate = true)
public class PageCreatedTask implements EventListener {
    private static final Logger log =
        LoggerFactory.getLogger(PageCreatedTask.class);
```

```
@Reference
private WorkflowService workflowService;
```

```
@Reference
private ResourceResolverFactory resolverFactory;
```

```
@Reference
private SlingRepository repository;
```

```
private Session observationSession;
```

```
@Activate
protected void activate() {
    try {
        observationSession = repository.loginService("hemanth", null);
```

```
observationSession.getWorkspace().getObservationManager().addEventListener(
    this,
    Event.NODE_ADDED,
    "/content", // or any specific path you want to observe
    true,
    null,
    null,
    false
);
```

```

        log.info("PageCreationEventListener activated");
    } catch (Exception e) {
        log.error("Error activating PageCreationEventListener", e);
    }
}

```

@Deactivate

```

protected void deactivate() {
    if (observationSession != null) {
        try {

```

```

        observationSession.getWorkspace().getObservationManager().removeEventListener(this);

```

```

            observationSession.logout();
        } catch (Exception e) {
            log.error("Error deactivating PageCreationEventListener", e);
        }
    }
}

```

@Override

```

public void onEvent(EventIterator events) {
    while (events.hasNext()) {
        Event event = events.nextEvent();
        try {
            String path = event.getPath();
            handleResourceAddedEvent(path);
        } catch (Exception e) {
            log.error("Error handling event", e);
        }
    }
}

```

```

private void handleResourceAddedEvent(String path) {
    ResourceResolver resourceResolver = null;
    try {
        Map<String, Object> param = new HashMap<>();

```

```

param.put(ResourceResolverFactory.SUBSERVICE, "hemanth");
resourceResolver = resolverFactory.getServiceResourceResolver(param);
Resource resource = resourceResolver.getResource(path);

if (resource != null && resource.isResourceType("cq:Page")) {
    log.info("Page created at path: {}", path);
    startWorkflow(path);
}
} catch (LoginException e) {
    log.error("Error obtaining resource resolver", e);
} finally {
    if (resourceResolver != null) {
        resourceResolver.close();
    }
}
}
}

```

```

private void startWorkflow(String payloadPath) {
    Map<String, Object> param = new HashMap<>();
    param.put(ResourceResolverFactory.SUBSERVICE, "hemanth");
    try (ResourceResolver resolver =
resolverFactory.getServiceResourceResolver(param)) {
        Session session = resolver.adaptTo(Session.class);
        WorkflowSession workflowSession =
workflowService.getWorkflowSession(session);

        String workflowModelPath = "/var/workflow/models/HemanthModel";
        WorkflowModel workflowModel =
workflowSession.getModel(workflowModelPath);
        WorkflowData workflowData =
workflowSession.newWorkflowData("JCR_PATH", payloadPath);

        workflowSession.startWorkflow(workflowModel, workflowData);
        log.info("Workflow started for payload: {}", payloadPath);

    } catch (LoginException e) {
        log.error("LoginException while starting workflow", e);
    }
}

```

```

    } catch (WorkflowException e) {
        log.error("WorkflowException while starting workflow", e);
    } catch (Exception e) {
        log.error("Exception while starting workflow", e);
    }
}
}
}

```

ADD PROPERTY THROUGH WORKFLOW

```
package workflow;
```

```
import java.util.Calendar;
import java.util.HashMap;
import java.util.Map;
```

```
import javax.jcr.Node;
import javax.jcr.Session;
```

```
import org.apache.sling.api.resource.ResourceResolver;
import org.apache.sling.api.resource.ResourceResolverFactory;
import org.osgi.service.component.annotations.Component;
import org.osgi.service.component.annotations.Reference;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
```

```
import com.adobe.granite.workflow.WorkflowSession;
import com.adobe.granite.workflow.exec.WorkItem;
import com.adobe.granite.workflow.exec.WorkflowProcess;
import com.adobe.granite.workflow.metadata.MetadataMap;
```

```
@Component(service = WorkflowProcess.class, property = {"process.label=Page
Created"})
```

```
public class PagecreatedWorkflow implements WorkflowProcess {
    private static final Logger log =
    LoggerFactory.getLogger(PagecreatedWorkflow.class);
```

```
@Reference
```

```

private ResourceResolverFactory resolverFactory;

@Override
public void execute(WorkItem workItem, WorkflowSession workflowSession,
MetadataMap metaDataMap) {
    String payloadPath = workItem.getWorkflowData().getPayload().toString();
    ResourceResolver resourceResolver = null;

    try {
        // Using service user to obtain resource resolver
        Map<String, Object> param = new HashMap<>();
        param.put(ResourceResolverFactory.SUBSERVICE, "hemanth");
        resourceResolver = resolverFactory.getServiceResourceResolver(param);

        Session session = resourceResolver.adaptTo(Session.class);
        Node pageNode = session.getNode(payloadPath);

        if (pageNode != null && pageNode.hasNode("jcr:content")) {
            Node contentNode = pageNode.getNode("jcr:content");
            String templatePath =
contentNode.getProperty("cq:template").getString();

            Calendar currentDate = Calendar.getInstance();

            if (!"/conf/Demo/settings/wcm/templates/page-
content".equals(templatePath)) {
                currentDate.add(Calendar.DATE, -1);
            }

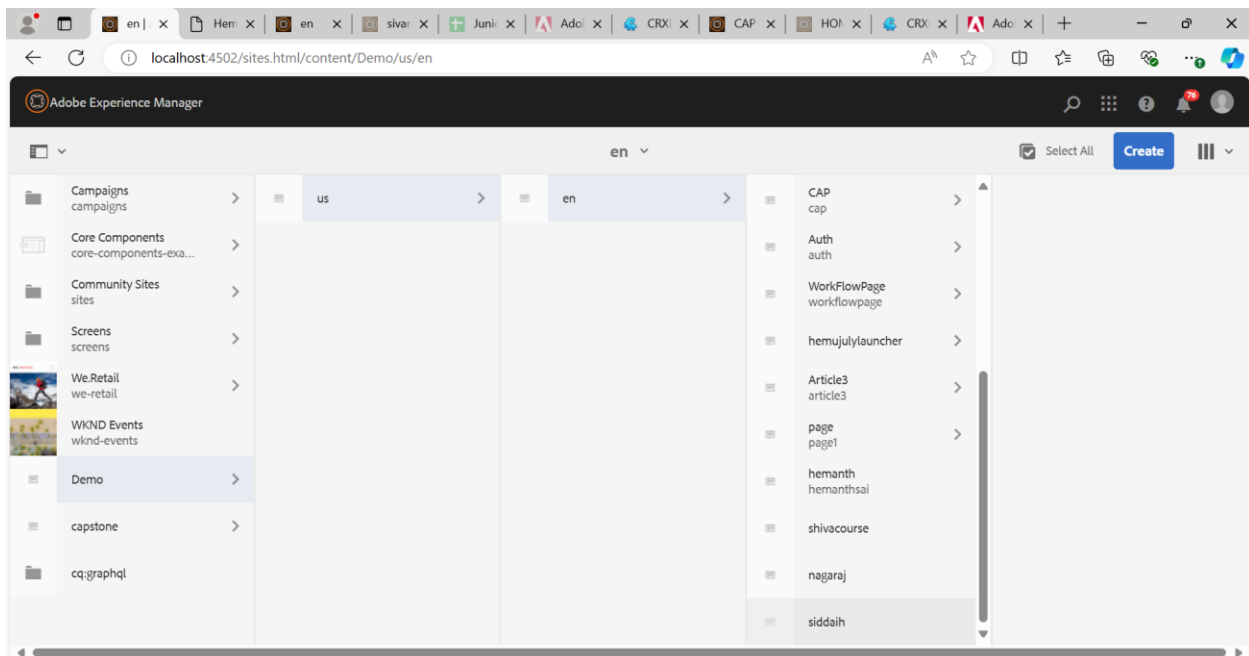
            contentNode.setProperty("expirydate", currentDate);
            session.save();
            log.info("Property 'expired' added to jcr:content of page at {}",
payloadPath);
        } else {
            log.warn("jcr:content node not found for page at {}", payloadPath);
        }
    } catch (Exception e) {

```

```

    log.error("Error adding property to jcr:content of page at {}", payloadPath,
e);
    throw new RuntimeException(e);
} finally {
    if (resourceResolver != null) {
        resourceResolver.close();
    }
}
}
}
}
}

```



```
C:\Users\Devi\Desktop\AEM\Author\crx-quickstart\logs\workflow.log - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
wknd3.log new 17 new 18 new 19 new 20 June.log error.log workflow.log new 21
181 apache.jackrabbit.oak-segment-tar:1.22.13]
182 rabbit.oak-segment-tar:1.22.13]
183 rabbit.oak-segment-tar:1.22.13]
184 ate.java:651) [org.apache.jackrabbit.oak-segment-tar:1.22.13]
185 jackrabbit.oak-store-spi:1.22.13]
186 pache.jackrabbit.oak-store-spi:1.22.13]
187 [org.apache.jackrabbit.oak-store-spi:1.22.13]
188 [org.apache.jackrabbit.oak-core:1.22.13]
189 [org.apache.jackrabbit.oak-store-spi:1.22.13]
190 [org.apache.jackrabbit.oak-store-spi:1.22.13]
191 ckrabbit.oak-segment-tar:1.22.13]
192 r.java:299) [org.apache.jackrabbit.oak-segment-tar:1.22.13]
193 er.java:270) [org.apache.jackrabbit.oak-segment-tar:1.22.13]
194 .apache.jackrabbit.oak-segment-tar:1.22.13]
195 rabbit.oak-core:1.22.13]
196 [org.apache.jackrabbit.oak-jcr:1.22.13]
197 rg.apache.jackrabbit.oak-jcr:1.22.13]
198
199 Model_913:/content/Demo/us/en/siddaih] workflow.PagecreatedWorkflow Property 'expired' added to jcr:content of page at /content/Demo/us/en/siddaih
200
```

localhost:4502/crx/de/index.jsp#/content/Demo/us/en/siddaih/jcr%3Acontent

CRXDE | Lite
Content Repository Extreme Development Environment Lite

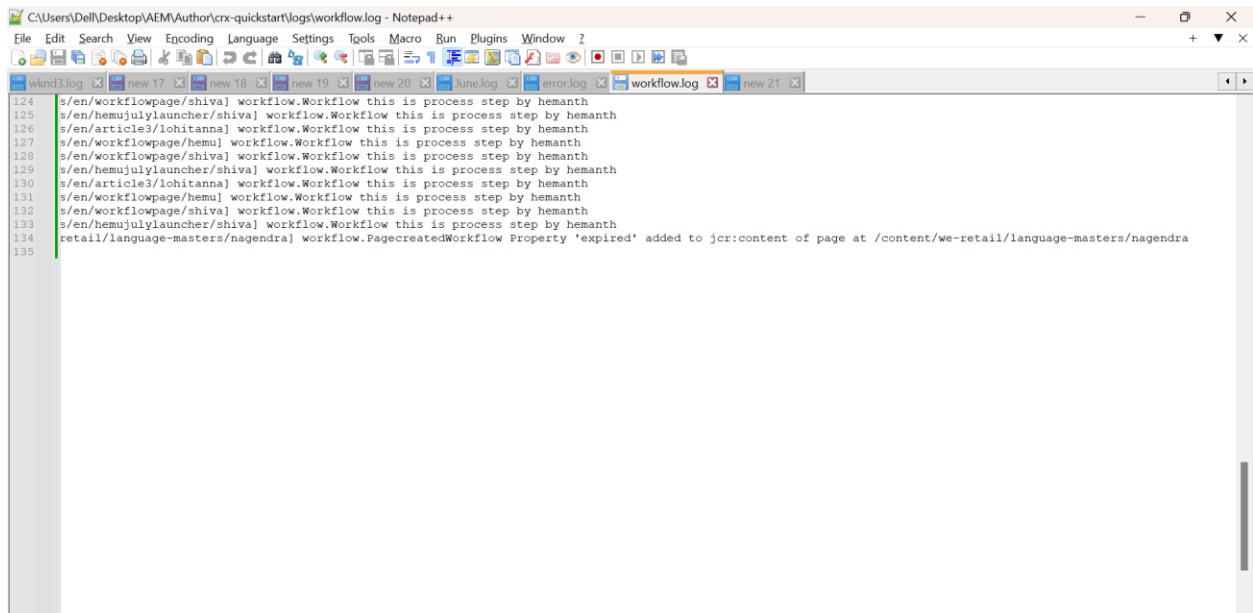
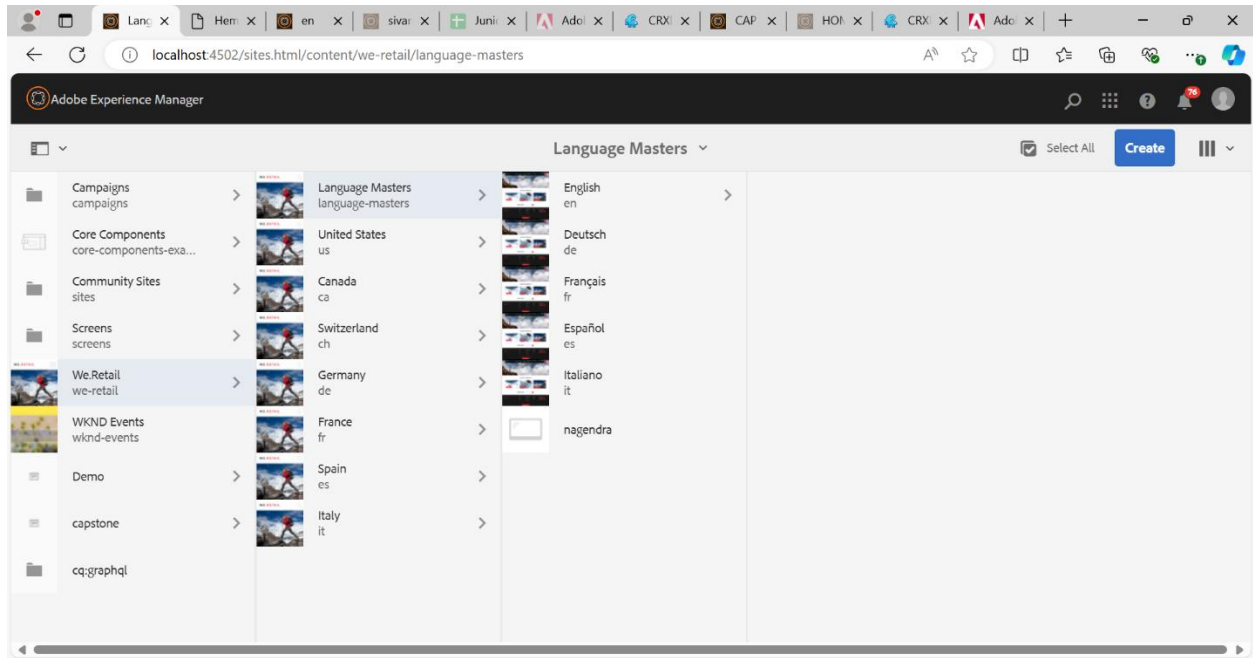
Enter search term to search the repository

Repository Information
Apache Jackrabbit Oak 1.7
The Apache Software Foundation

Developer Resources

Name	Type	Value	Protected	Mandatory	Multiple	Auto Created
5 cq:lastReplicationAction	String	Activate	false	false	false	false
6 cq:template	String	/content/Demo/settings/wcm/templates/page-c...	false	false	false	false
7 expirydate	Date	2024-07-23T16:22:36.563+05:30	false	false	false	false
8 jcr:baseVersion	Reference	e4a56a2f-5e13-42f7-b7cb-9c2e36b6a0ad	true	true	false	false
9 jcr:created	Date	2024-07-23T16:22:34.193+05:30	true	false	false	true
10 jcr:createdBy	String	admin	true	false	false	true
11 jcr:isCheckedOut	Boolean	true	true	true	false	true

Name: Type: String Multi Add Clear



localhost4502/crx/de/index.jsp#/content/we-retail/language-masters/nagendra/jcr%3Acontent

catalogs

screens

community

community-components

we-retail

jcr:content

language-masters

jcr:content

en

de

fr

es

it

nagendra

jcr:content

root

us

ca

ch

de

jcr:content

de

screens3

jcr:content

root

fr

es

Home

coursecontent.html

Course Detail Component.ht

coursecontent.html

1<!DOCTYPE html>

2<html lang="en">

3<head>

4<meta charset="UTF-8">

5<meta names="viewport" content="width=device-width, initial-scale=1.0">

6<title>AEM Topics</title>

7<link rel="stylesheet" href="styles.css"> <!-- Link to your external CSS file -->

8<style>

9/* Basic styling for layout */

10body {

11font-family: Arial, sans-serif;

12line-height: 1.6;

13background-color: #f2f2f2;

14padding: 20px;

15}

Properties

Access Control

Replication

Console

Build Info

Name	Type	Value	Protected	Mandatory
5 jcr:created	Date	2024-07-23T16:20:15.743+05:30	true	false
6 expirydate	Date	2024-07-22T16:20:22.682+05:30	false	false
7 cq:xfVariantType	String	web	false	false
8 cq:xfShowInEditor	Boolean	true	false	false
9 cq:template	String	/conf/we-retail/settings/wcm/templates/expe...	false	false
10 cq:lastModifiedBy	String	admin	false	false
11 cq:lastModified	Date	2024-07-23T16:20:15.749+05:30	false	false

Name

Type

String

Multi

Add