Date: 25-03-25

Creating a Custom Workflow in AEM

A custom workflow will be developed in Adobe Experience Manager (AEM) to handle specific tasks efficiently.

Steps to Create the Workflow Model:

1. Navigate to **Tools > Workflow > Models** in AEM.
2. Create a new workflow model and name it **"My Custom Workflow."**
3. Add workflow steps to define the processing logic.
4. Save and activate the workflow to enable its execution.

Implementing a Custom Workflow Process to Log Page Titles

A workflow process step will be created to extract and log page titles along with metadata.

Steps to Create the Workflow Process:

1. Develop a custom workflow process class.
2. Extract the page title from the resource.
3. Log the page title and relevant metadata in AEM logs.
4. Assign the workflow to a page and execute it

CustomWorkflowProcss.java

```
@Component(service = WorkflowProcess.class, immediate = true)
public class CustomWorkflowProcess implements WorkflowProcess {
   private static final Logger LOG = LoggerFactory.getLogger(CustomWorkflowProcess.class);
   @Override
   public void execute(WorkItem workItem, WorkflowSession workflowSession, MetaDataMap metaDataMap)
         throws WorkflowException {
       String pageTitle = workItem.getWorkflowData().getPayload().toString();
       LOG.info("Processing page: " + pageTitle);
   }
}
```

Creating an Event Handler to Log Resource Path in AEM

An event listener will be set up to detect changes in resources and log their paths.

Steps:

1. Develop an OSGi-based event listener.
2. Listen for events related to resource modifications, additions, or deletions.
3. Log the resource path in AEM logs.

EventHandler.java

```
@Component(service = EventHandler.class)
public class CustomEventHandler implements EventHandler {
  private static final Logger LOG = LoggerFactory.getLogger(CustomEventHandler.class);
  @Override
  public void handleEvent(Event event) {
    LOG.info("Resource changed: " + event.getProperty("path"));
  }
}
```

Scheduling a Sling Job to Log "Hello World"

A Sling Job will be created to log a simple message when triggered.

Steps:

1. Develop a Sling Job class.
2. Log the "Hello World" message when executed.
3. Trigger the job manually or programmatically.

SlingJob.java

```
@Component(service = JobConsumer.class, immediate = true,
     property = {JobConsumer.PROPERTY_TOPICS + "=com/example/hello"})
public class HelloWorldJob implements JobConsumer {
  private static final Logger LOG = LoggerFactory.getLogger(HelloWorldJob.class);
  @Override
  public JobResult process(Job job) {
    LOG.info("Hello World from Sling Job");
    return JobResult.OK;
  }
}
```

Setting Up a Scheduler to Log "Hello World" Every 5 Minutes

An AEM scheduler will be configured to run at regular intervals.

Steps:

1. Develop an OSGi-based scheduler.
2. Set up a cron expression to run every 5 minutes.
3. Log the "Hello World" message upon execution.

Scheduler.java

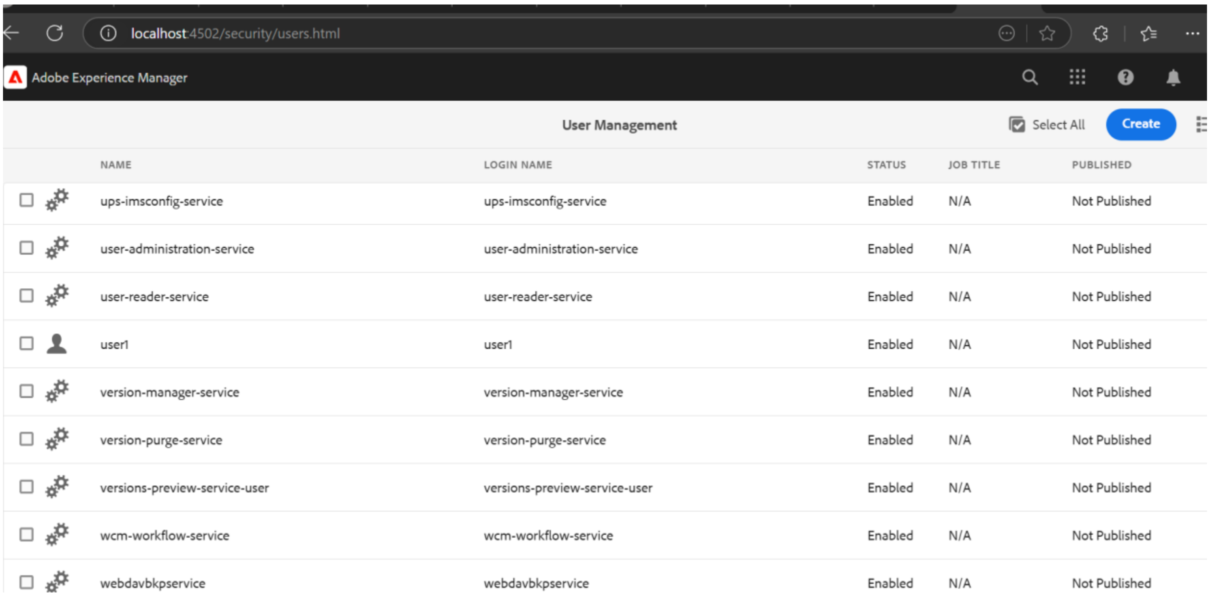
```
@Component(service = Runnable.class, immediate = true,
     property = {"scheduler.expression=0 */5 * * * ?"})
public class HelloWorldScheduler implements Runnable {
  private static final Logger LOG = LoggerFactory.getLogger(HelloWorldScheduler.class);
  @Override
  public void run() {
    LOG.info("Hello World");
  }
}
```

Creating Users and Assigning Permissions in AEM

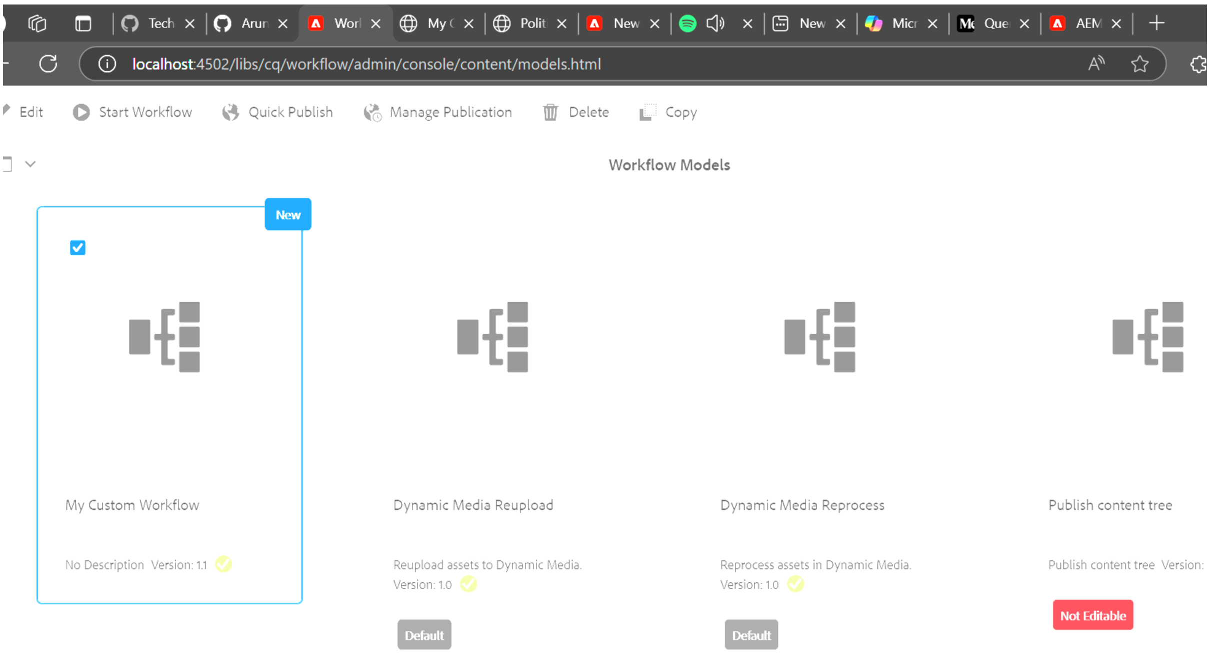
Three users will be created and added to a group with specific access rights.

Steps:

1. Go to **Tools > Security > Users** in AEM.
2. Create three new users: **user1, user2, user3**.
3. Navigate to **Groups** and create a new group named **Dev Author**.
4. Add all three users to the **Dev Author** group.
5. Grant **read-only** access to /content and /dam.
6. Provide **replication access** to the group.



	NAME	LOGIN NAME	STATUS	JOB TITLE	PUBLISHED
□	ups-imsconfig-service	ups-imsconfig-service	Enabled	N/A	Not Published
□	user-administration-service	user-administration-service	Enabled	N/A	Not Published
□	user-reader-service	user-reader-service	Enabled	N/A	Not Published
□	user1	user1	Enabled	N/A	Not Published
□	version-manager-service	version-manager-service	Enabled	N/A	Not Published
□	version-purge-service	version-purge-service	Enabled	N/A	Not Published
□	versions-preview-service-user	versions-preview-service-user	Enabled	N/A	Not Published
□	wcm-workflow-service	wcm-workflow-service	Enabled	N/A	Not Published
□	webdavbkpservice	webdavbkpservice	Enabled	N/A	Not Published



☑ My Custom Workflow

No Description Version: 1.1

Dynamic Media Reupload

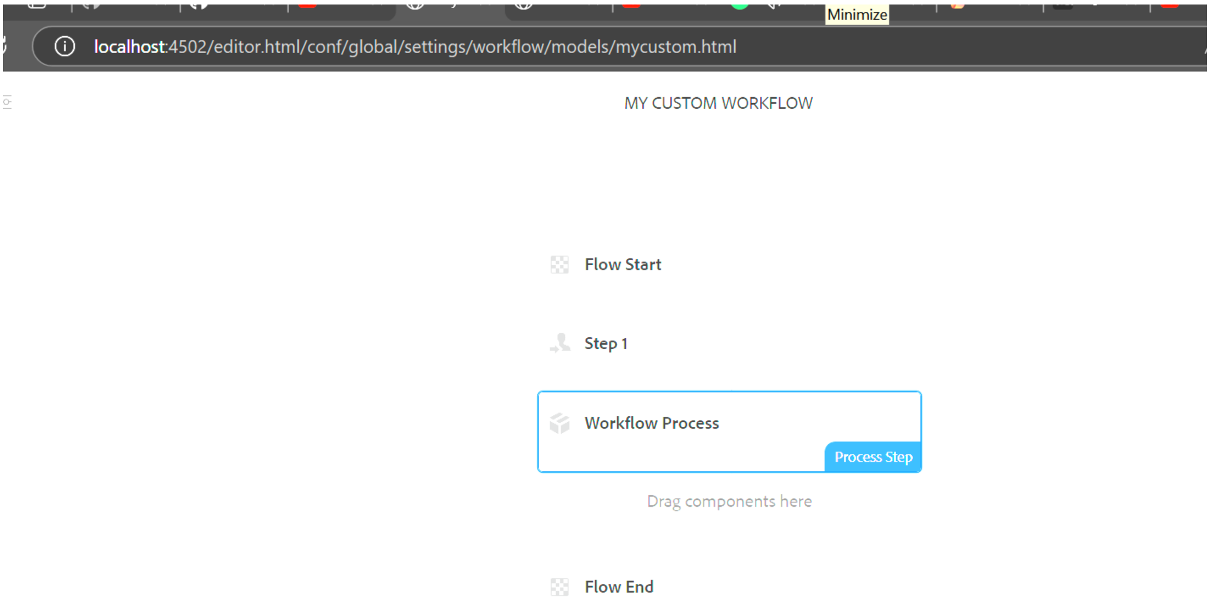
Reupload assets to Dynamic Media. Version: 1.0

Dynamic Media Reprocess

Reprocess assets in Dynamic Media. Version: 1.0

Publish content tree

Publish content tree Version:



localhost:4502/editor.html/conf/global/settings/workflow/models/mycustom.html