

## Create a Custom Workflow in AEM

### Description

Develop a custom workflow in AEM to handle specific content processing tasks.

### Steps

1. **Open Workflow Models**
  - Navigate to **Tools** → **Workflow** → **Models** in AEM.
2. **Create a New Workflow Model**
  - Click **Create** → **Create Workflow Model**.
  - Enter the **Title** as "My Custom Workflow".
  - Click **Done** to save the model.
3. **Add Workflow Steps**
  - Open the newly created workflow model.
  - Drag and drop required workflow **process steps** from the **Side Panel**.
  - Configure each step as needed.
4. **Save & Activate the Workflow**
  - Click **Save** to store changes.
  - Click **Sync** to activate the workflow for use.

## Create a Custom Workflow Process to Log Page Titles in AEM

### Description

Develop a custom **AEM Workflow Process step** to log page titles and metadata when a workflow runs on a page.

### Steps

1. **Create a Custom Workflow Process Class**
  - Develop a new Java class that implements WorkflowProcess.
2. **Extract the Page Title from the Resource**
  - Retrieve the page payload from the workItem and extract the title.
3. **Log the Title and Metadata**
  - Use the AEM logger to print the page title and other metadata.
4. **Assign & Run the Workflow**
  - Attach the workflow to a page and execute it.

### Code Snippet (CustomWorkflowProcess.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);

        }

}
```

### 1. Create an Event Handler to Log Resource Path

**Description:** Log resource changes (added, modified, deleted) in AEM.

#### Steps:

- Create an **OSGi Event Listener**.
- Listen for **resource changes**.
- Log the **resource path**.

#### Code Snippet:

```
@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"));

    }

}
```

## 2. Create a Sling Job to Log "Hello World"

**Description:** A Sling Job that logs a message when triggered.

**Steps:**

- Create a **Sling Job class**.
- Log **"Hello World"**.
- Trigger manually or programmatically.

**Code Snippet:**

```
@Component(service = JobConsumer.class, immediate = true,
    property = {JobConsumer.PROPERTY_TOPICS + "=com/gautam/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;
    }
}
```

## 3. Create a Scheduler to Log "Hello World" Every 5 Minutes

**Description:** Schedule a log message every **5 minutes** using a cron job.

**Steps:**

- Implement **OSGi Scheduler**.
- Set **cron expression** (0 \*/5 \* \* \* ?).
- Log **"Hello World"**.

**Code Snippet:**

```
@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");
}
}

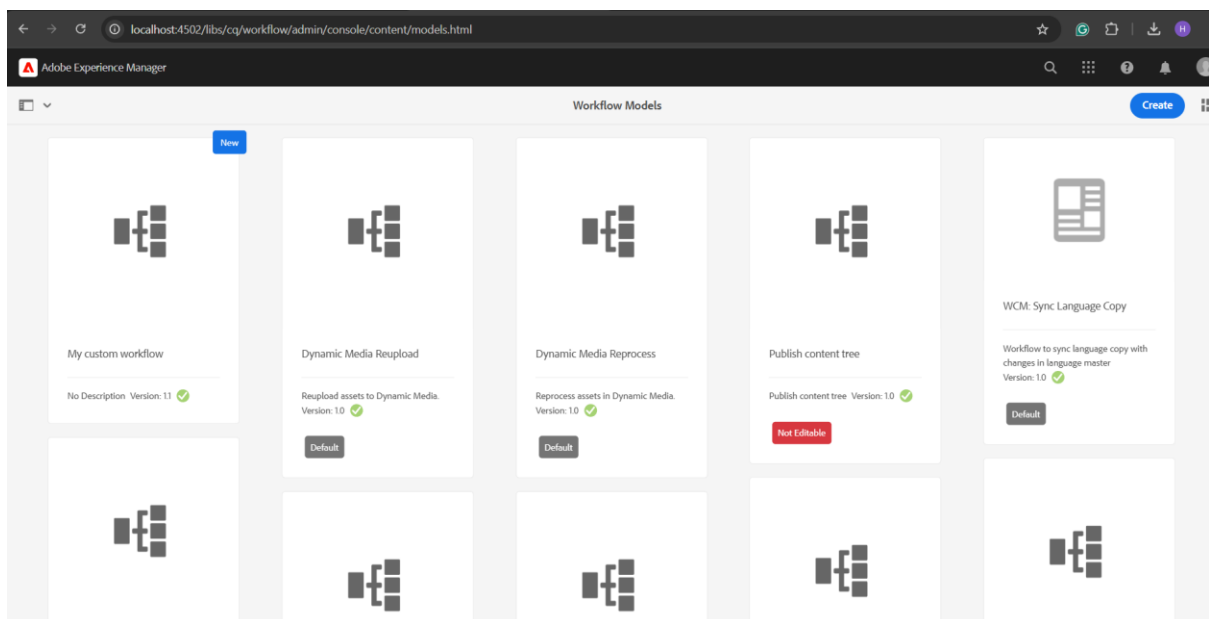
```

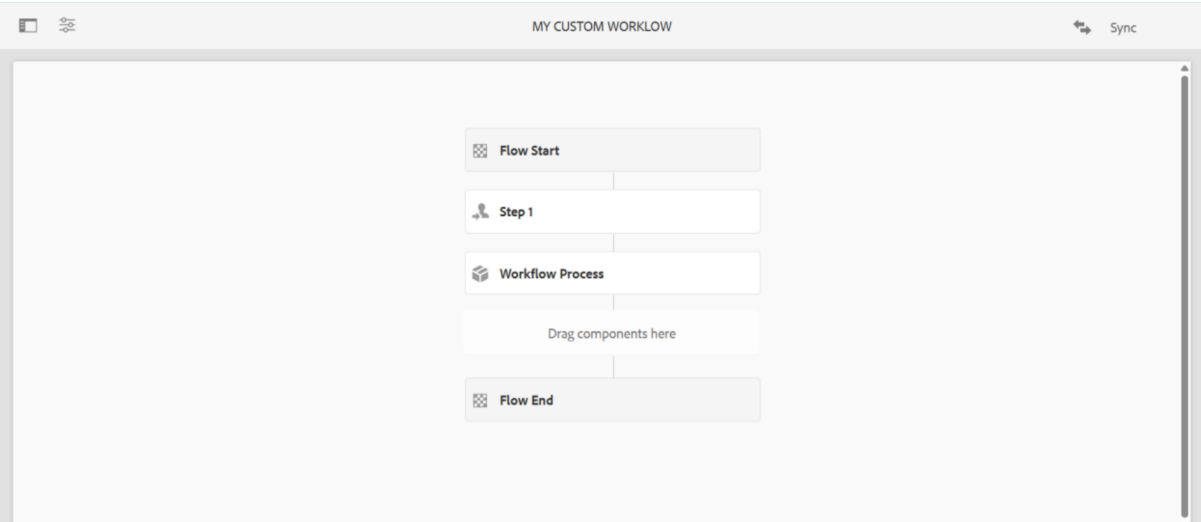
## 4. Create Users and Assign Permissions

**Description:** Create users and assign **read + replication** access.

**Steps:**

- Go to **Tools > Security > Users**.
- Create **user1, user2, user3**.
- Create **Dev Author** group.
- Assign **read-only** access to **/content & /dam**.
- Grant **replication** permission.





Adobe Experience Manager

User Management

Select All Create

	NAME	LOGIN NAME	STATUS	JOB TITLE	PUBLISHED
<input type="checkbox"/> ⚙️	user-reader-service	user-reader-service	Enabled	N/A	Not Published
<input type="checkbox"/> 👤	user1	user1	Enabled	N/A	Not Published
<input type="checkbox"/> 👤	user2	user2	Enabled	N/A	Not Published
<input type="checkbox"/> 👤	user3	user3	Enabled	N/A	Not Published

Edit Group Settings For Dev Authors

Cancel Save & Close

Activate Deactivate

Details Members

**Add Members to this Group**

Select User or Group

Type User or Group Name

**Group Members**

👤	user1	×
👤	user2	×
👤	user3	×