

## AEM Custom Workflow, Event Handler, Sling Job, and Scheduler Implementation

### 1. Create Custom Workflow

- Create a workflow named "My Custom Workflow".
- Configure it to be triggered on page modifications or activations.

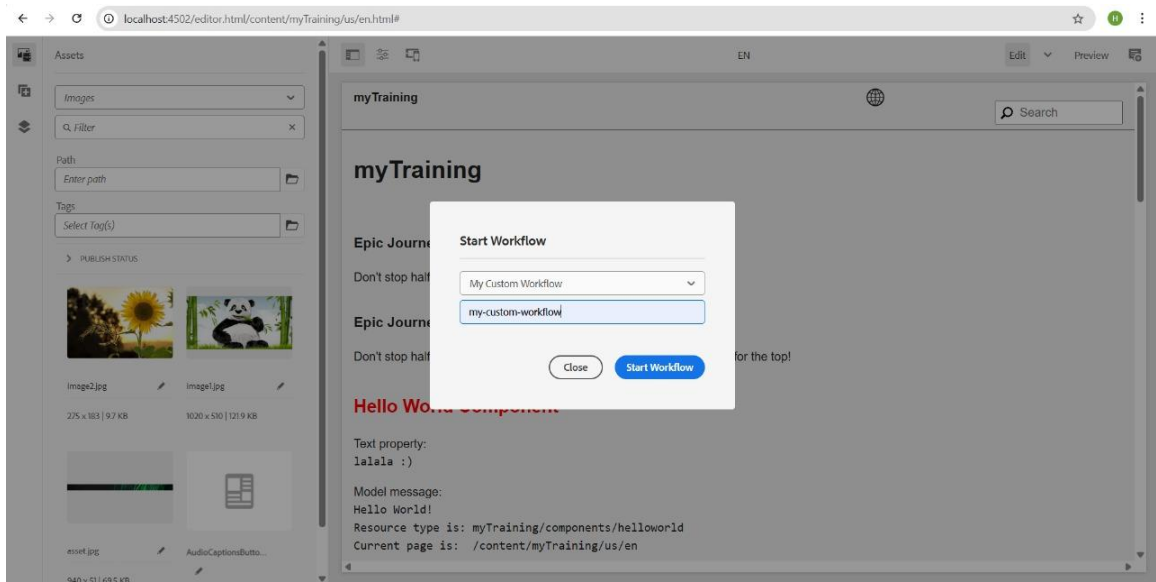
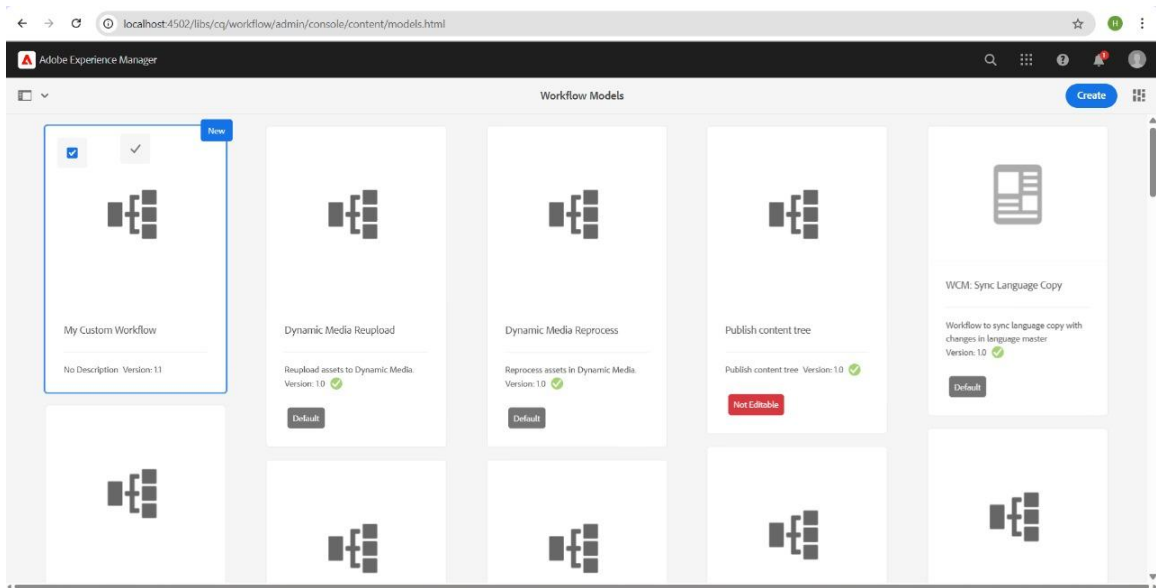
### 2. Create Custom Workflow Process

- Implement a Workflow Process Step that:
  - Retrieves the Page Title from the resource.
  - Prints it in the logs.
- Deploy and run the workflow on a page to verify metadata logging.

#### Custom Workflow Process Implementation:

```
@Component(service = WorkflowProcess.class, property =
{ WorkflowConstants.PROCESS_LABEL + "= Custom Page Title Logger" })
public class CustomWorkflowProcess implements WorkflowProcess {
    private static final Logger LOG =
LoggerFactory.getLogger(CustomWorkflowProcess.class);

    @Override
    public void execute(WorkItem workItem, WorkflowSession workflowSession,
MetaDataMap args) throws WorkflowException {
        String pagePath = workItem.getWorkflowData().getPayload().toString();
        ResourceResolver resourceResolver =
workflowSession.adaptTo(ResourceResolver.class);
        Resource resource = resourceResolver.getResource(pagePath + "/jcr:content");
        if (resource != null) {
            ValueMap properties = resource.getValueMap();
            String pageTitle = properties.getOrDefault("jcr:title", "No Title").toString();
            LOG.info("Page Title: { }", pageTitle);
        }
    }
}
```



### 3. Create Event Handler in AEM

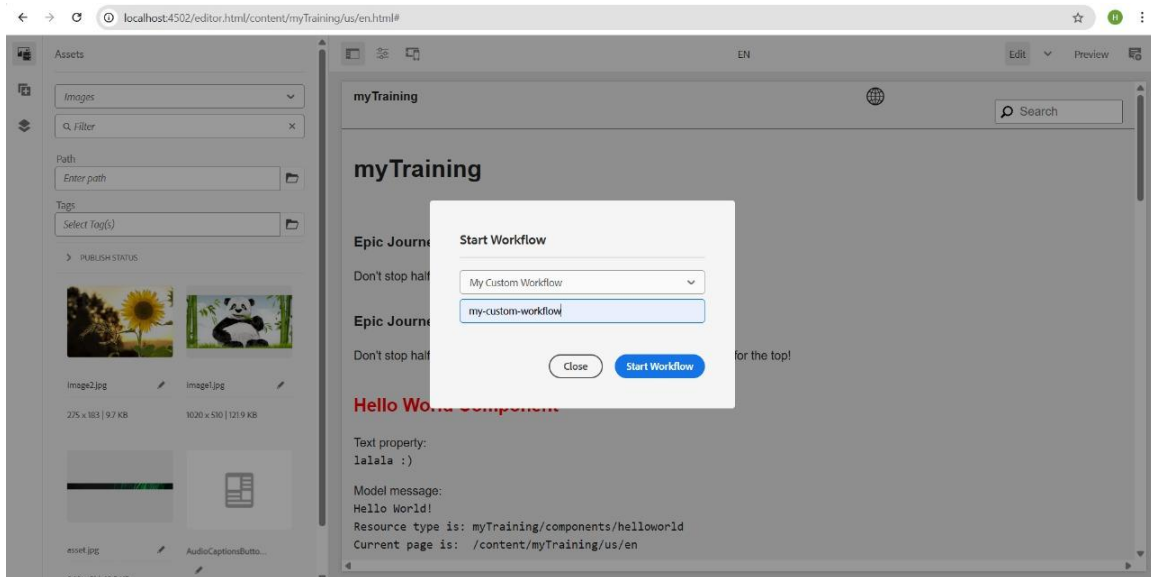
- Implement an Event Handler that listens for modifications in /content and /dam.
- Log the resource path when an event occurs.

#### Event Handler Implementation:

```
@Component(immediate = true, service = EventHandler.class,
    property = { EventConstants.EVENT_TOPIC +
        "=org/apache/sling/api/resource/Resource/ADDED" })
public class CustomEventHandler implements EventHandler {
```

```
private static final Logger LOG =
LoggerFactory.getLogger(CustomEventHandler.class);
```

```
@Override
public void handleEvent(Event event) {
    String resourcePath = (String)
event.getProperty(SlingConstants.PROPERTY_PATH);
    LOG.info("Resource added at: {}", resourcePath);
}
}
```



#### 4. Create Sling Job to Print "Hello World" in Logs

##### Sling Job Implementation:

```
@Component(service = JobConsumer.class, property =
{JobConsumer.PROPERTY_TOPICS + "=my/custom/job"})
public class HelloWorldSlingJob implements JobConsumer {
    private static final Logger LOG =
LoggerFactory.getLogger(HelloWorldSlingJob.class);
```

```
@Override
public JobResult process(Job job) {
    LOG.info("Hello World from Sling Job");
    return JobResult.OK;
}
}
```

```
}
```

## 5. Create Scheduler to Print "Yellow World" Every 5 Minutes

- Use a cron expression (`0 0/5 \* \* \* ?`) for scheduling.

### Scheduler Implementation:

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

## 6. Create 3 Users, Add to Group, and Set Permissions

1. Create a new group: "Dev Authors".
2. Add three users to this group.
3. Assign Read-Only access to `/content` and `/dam`.
4. Grant Replication Access to the group.

### Permission Settings:

- Go to User Administration in AEM.
- Create a group "Dev Authors".
- Add users to the group.
- Navigate to Permissions and apply:
  - Read Only for `/content` and `/dam`.
  - Replication Rights under Permissions & Policies.

**Deployment & Testing:**

1. Deploy and test the Custom Workflow on pages.
2. Validate Workflow Process Step logging page titles.
3. Trigger Event Handler and check logs for resource path.
4. Manually run Sling Job to log "Hello World".
5. Confirm Scheduler logs "Yellow World" every 5 minutes.
6. Ensure users in Dev Authors group have correct access.