AEM Assignment

**DATE : 18-03-2025**

**1. Maven Lifecycle**

Maven is a build automation tool used primarily for Java projects. It follows a predefined lifecycle to manage project build processes. The three key lifecycles are:

- Clean : Removes previous build artifacts.

- Default (Build) : Compiles, tests, packages, and installs the project.

- Site : Generates project documentation.

**Maven Build Phases**

1. validate – Checks if the project structure is correct.

2. compile – Compiles the source code.

3. test – Runs unit tests.

4. package – Packages the compiled code into a distributable format (e.g., JAR, WAR).

5. verify – Runs integration tests.

6. install – Installs the package into the local repository.

7. deploy – Deploys the final build to a remote repository.

**2. What is pom.xml and Why We Use It?**

The pom.xml (Project Object Model) file is the configuration file for Maven projects. It defines:

- Project information

- Dependencies

- Plugins

- Build settings

- Profiles

Example pom.xml structure:

xml

<project xmlns="http://maven.apache.org/POM/4.0.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0

http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.example</groupId>

<artifactId>sample-project</artifactId>

<version>1.0.0</version>

<dependencies>

<!-- Dependencies here -->

</dependencies>

</project>

**3. How Dependencies Work?**

Dependencies define external libraries required for a project. They are managed through pom.xml under the <dependencies> section.

Example:

xml

<dependency>

<groupId>org.apache.commons</groupId>

<artifactId>commons-lang3</artifactId>

<version>3.12.0</version>

</dependency>

Maven automatically downloads dependencies from repositories and places them in the target directory.

**4. Checking the Maven Repository**

Maven repositories store libraries and dependencies. The main types are:

- Local Repository (~/.m2/repository)

- Central Repository (https://repo.maven.apache.org/maven2/)

- Remote Repository (Configured in pom.xml for custom dependencies)

**5. How All Modules Build Using Maven**

In a multi-module project, the parent POM manages submodules. The pom.xml includes:

xml

<modules>

<module>ui.apps</module>

<module>ui.content</module>

<module>ui.frontend</module>

</modules>

Running mvn install at the root level builds all modules.

**6. Can We Build a Specific Module?**

Yes, using:

sh

mvn install -pl ui.apps -am

- -pl: Specifies the module to build.

- -am: Builds required dependencies.

**7. Role of ui.apps, ui.content, and ui.frontend**

- ui.apps : Stores AEM OSGi configurations and component code.

- ui.content : Stores content packages and templates.

- ui.frontend : Stores front-end resources like JavaScript, CSS, and React code.

**8. Why We Use Run Mode?**

Run modes configure different environments (e.g., development, production). AEM supports:

- author (Admin environment for content management)

- publish (User-facing site environment)

- dev, stage, prod (Custom modes for environments)

Set run modes in sling.properties or environment variables.

**9. What is Publish Environment?**

The publish environment serves live content to users. It is optimized for:

- Fast content delivery

- User access management

- Integration with the dispatcher

**10. Why We Use Dispatcher?**

AEM Dispatcher is a caching and load-balancing tool used to:

- Improve performance by caching pages.

- Protect AEM from high traffic loads.

- Restrict access based on security rules.

**11. From Where Can We Access crx/de?**

The CRX/DE (Content Repository eXtreme/Developer Edition) is accessible at:

http://localhost:4502/crx/de

It allows:

- Managing AEM JCR nodes.

- Editing repository content.

- Inspecting components and configurations.

**DATE**: 19/03/2025

1. **What is DAM and why we use it?**  
   DAM (Digital Asset Management) simplifies storing, organizing, and accessing digital assets, enhancing efficiency and collaboration in content workflows.
2. **Create a folder and upload images:**  
   Create the path /content/dam/myTraining/us/en-us, upload two images there, and use the AEM Image component to display them.
3. **What are renditions?**  
   Renditions are auto-generated versions of an image in various resolutions and formats, optimized for different display or usage scenarios.
4. **Add fields to Hello World component:**  
   Add FirstName and LastName fields in the component, retrieve their values via properties, and display them on hello world.html.
5. **@ValueMapValue annotation usage:**  
   In HelloWorldModel, use @ValueMapValue to auto-bind the FirstName and LastName properties from the JCR repository.
6. **Using Package Manager:**  
   Create two packages—one for the DAM images, another for the Hello World component—and use the replication agent to publish them from the author to the publish environment (4502 to 4503).

**DATE**: 20/03/2025

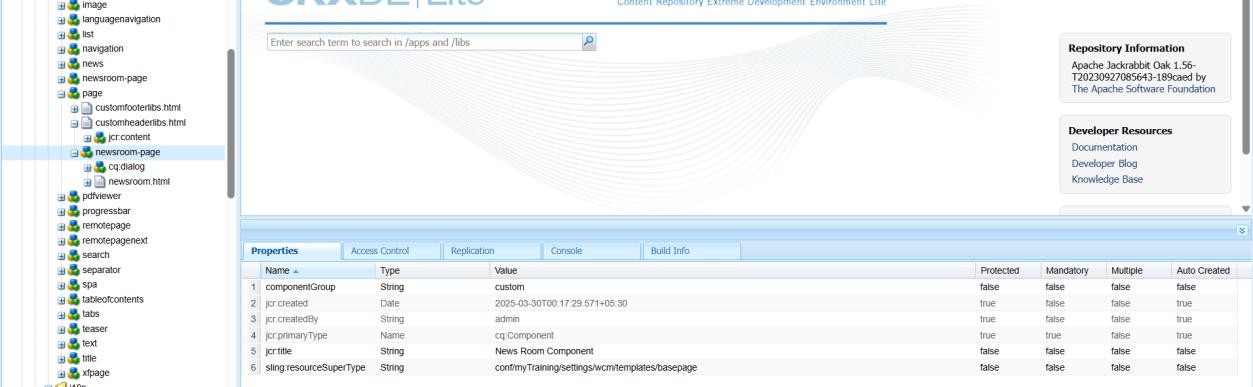
1. **Create news component:**  
   Define fields: title (h2), detail (p), published date, and source. Use Sling Model to map fields for rendering.
2. **Create multi-field news component:**  
   Design Sling Model fields for multiple news items: title and source. Add multifield functionality in the dialog for dynamic entry.
3. **Create Clientlibs:**  
   Define CSS/JS files under clientlibs folder. Create categories, include them in components using data-sly-resource.
4. **Apply colors:**  
   In CSS file, set:

* h2 { color: green; }
* p { color: yellow; }
* Date as color: black;.

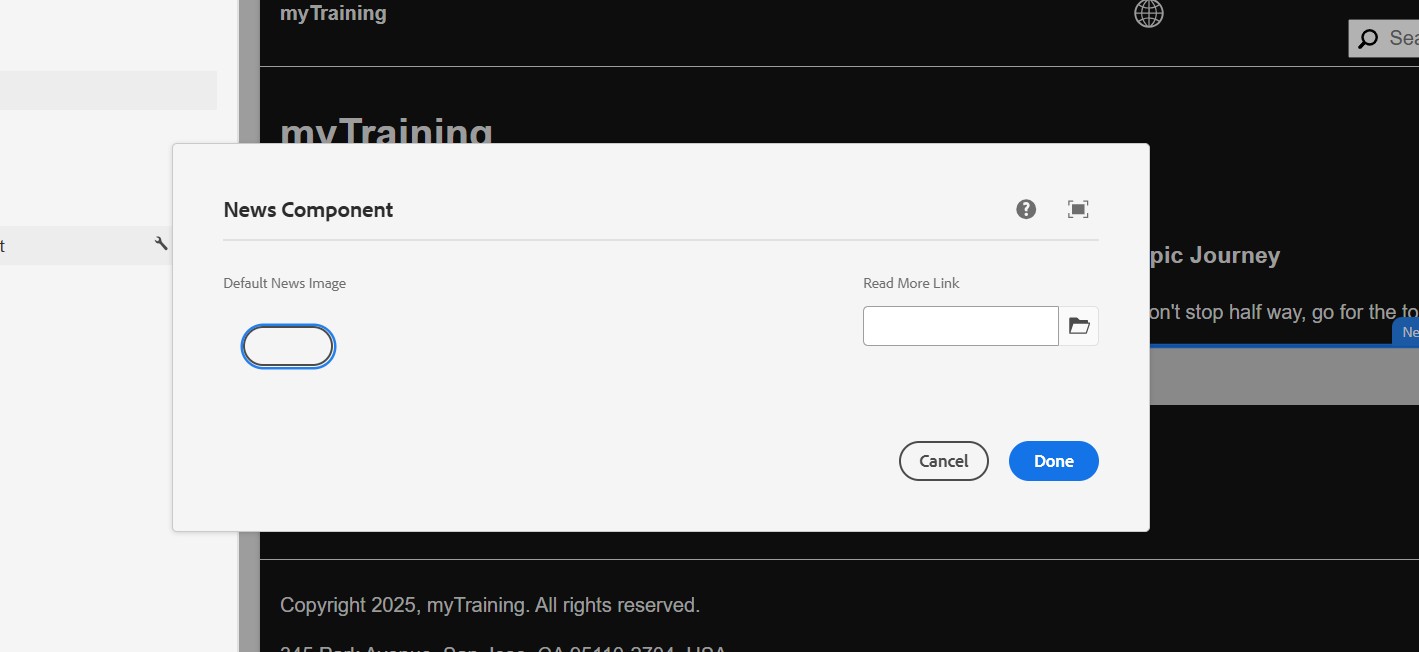
1. **Add component style:**  
   Include style attribute in dialog with a custom class name (cop-news-component) for applying unique styles to the news component.
2. **Base page metadata:**  
   Add new metadata file, set og:title, og:description, and og:image properties. Render them dynamically in base page.html.
3. **Custom page properties:**  
   Create fields for Global properties: og:title, og:description, and og:image path in page properties dialog and bind them.
4. **Extra Clientlibs:**  
   Additional client libraries (extraclientlibs) can be added for components. Include optional client-side resources improving component flexibility.

**DATE:** 21/03/2025

1. Create a News Room Page Component using the Base Page Component



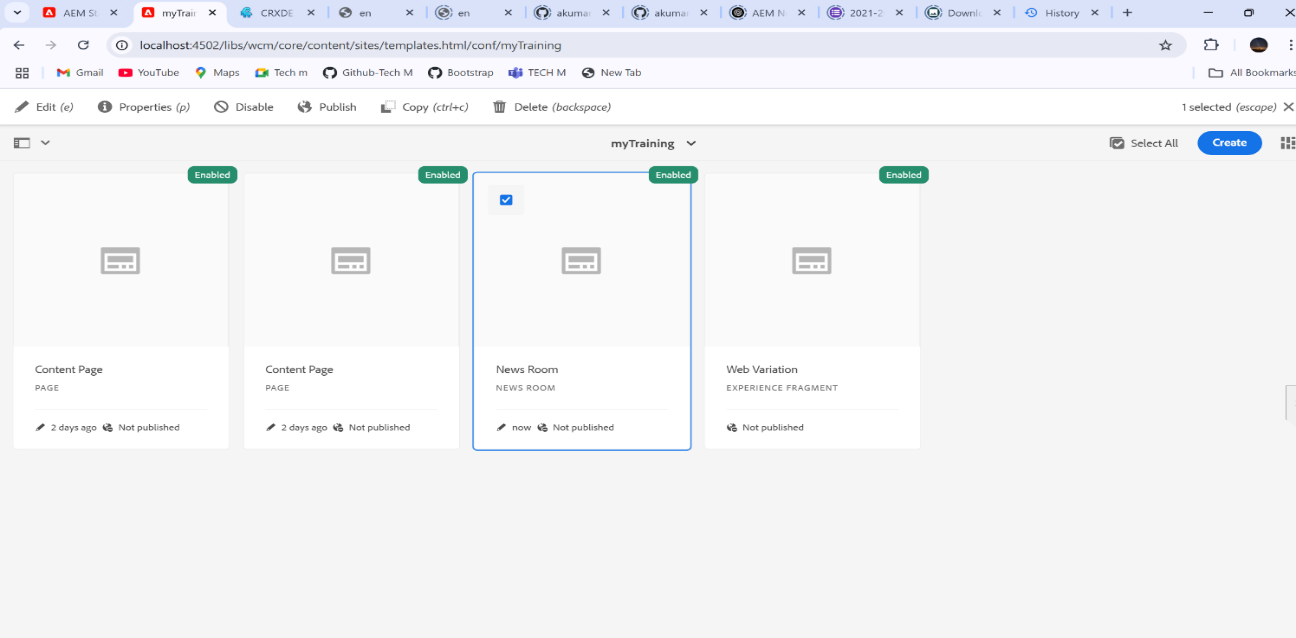
1. Create Custom Page Property: "NEWS Configurations"
2. Add "Read More" CTA for News Articles



1. **Create News Room Template Type using News Room Page Component**

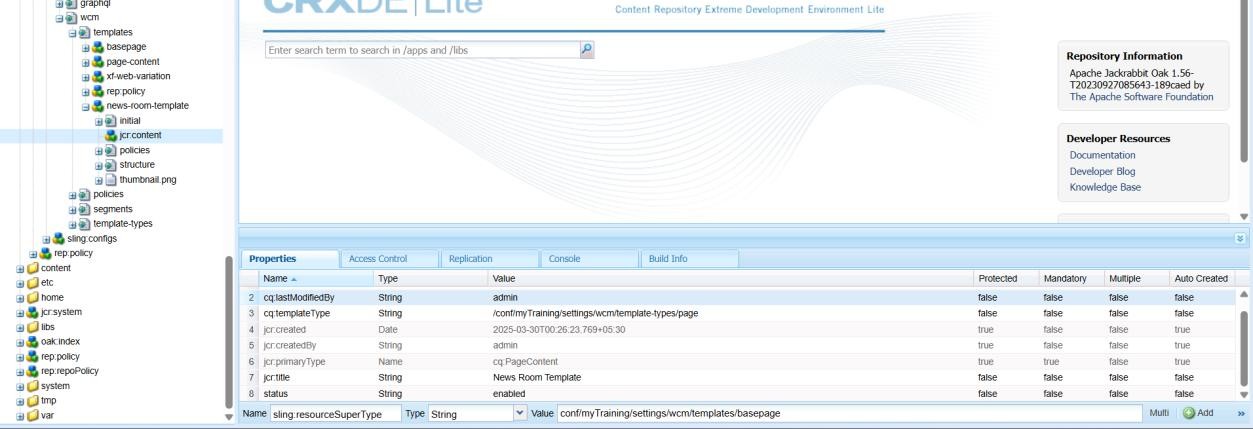
**Steps to Implement:**

* + Go to **AEM > Tools > General > Template Types**.
  + Create a new Template Type named **News Room Template Type**.
  + Set the **Allowed Resource Type** as your-project/components/page/newsroompage.



**How to Verify:**

* + Go to **Template Editor** and check if the template type appears.
  + Try creating a template using this type.



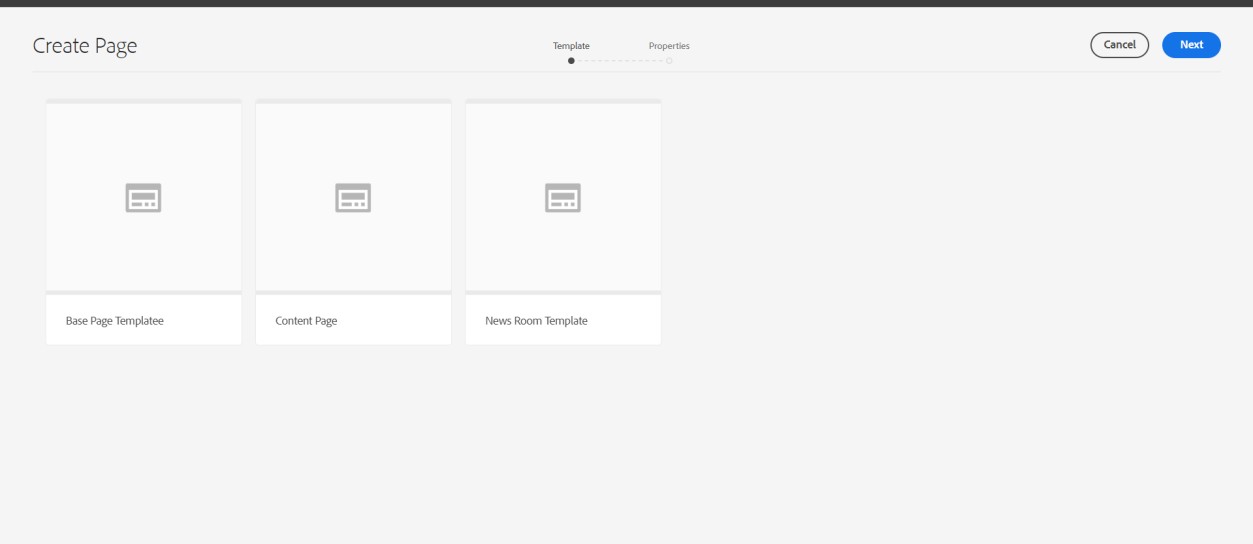
1. **Create News Room Template using News Room Template Type**

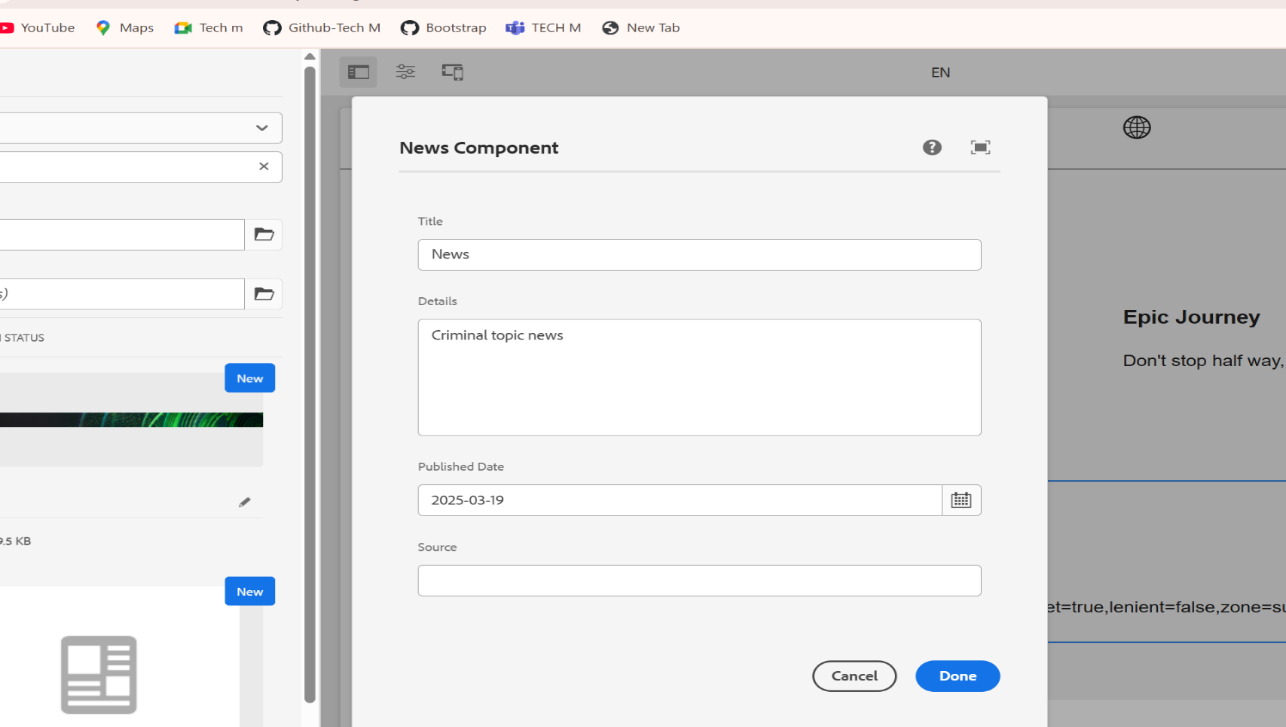
**Steps to Implement:**

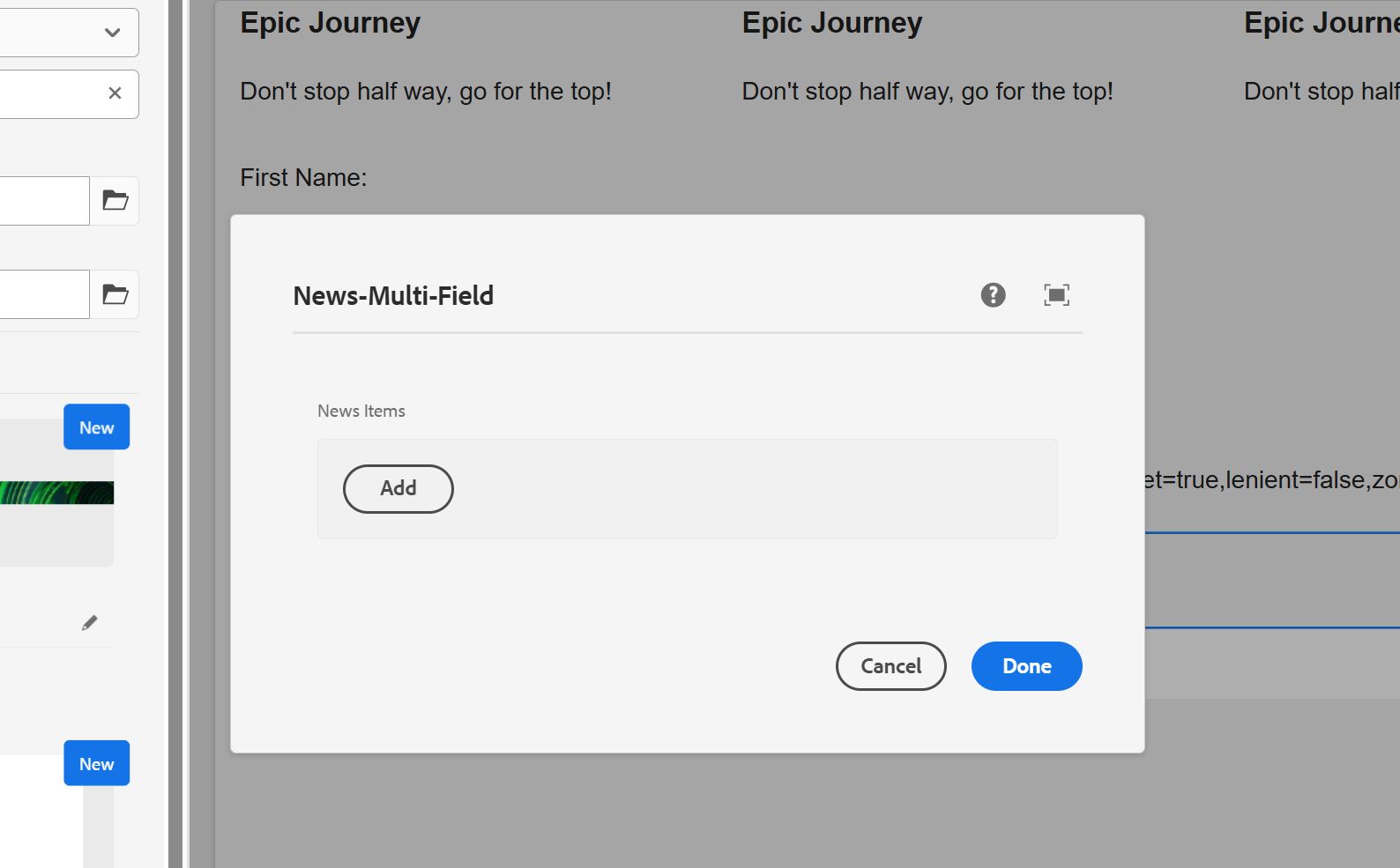
* + In **AEM > Templates**, create a new template using **News Room Template Type**.
  + Define the structure (Header, News Listing, Footer, etc.).

**How to Verify:**

* + Check if the template appears under **Templates** in AEM.
  + Create a page using this template and verify the structure.

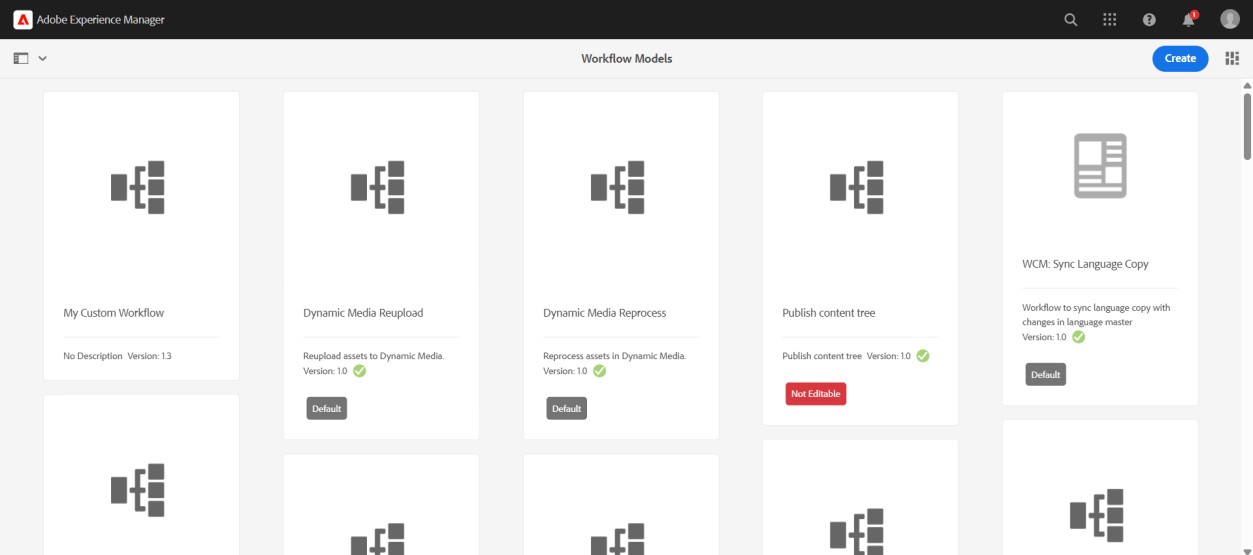


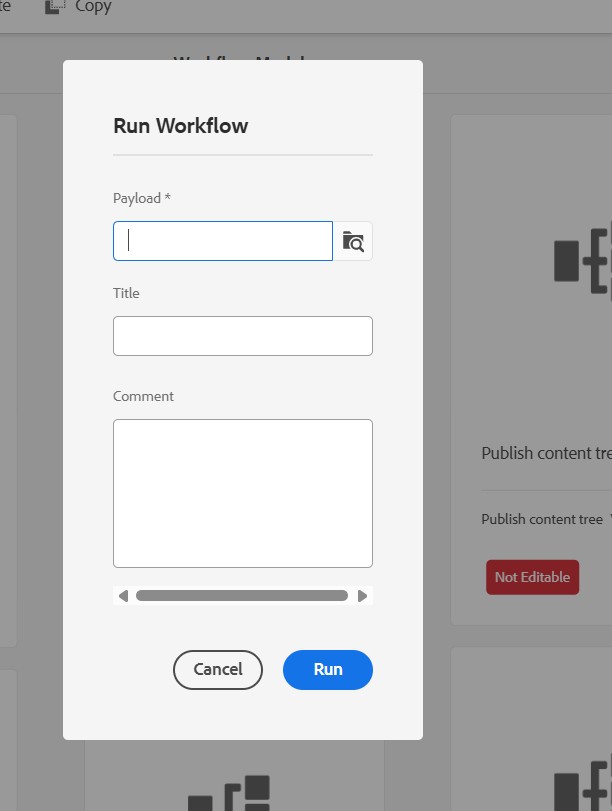


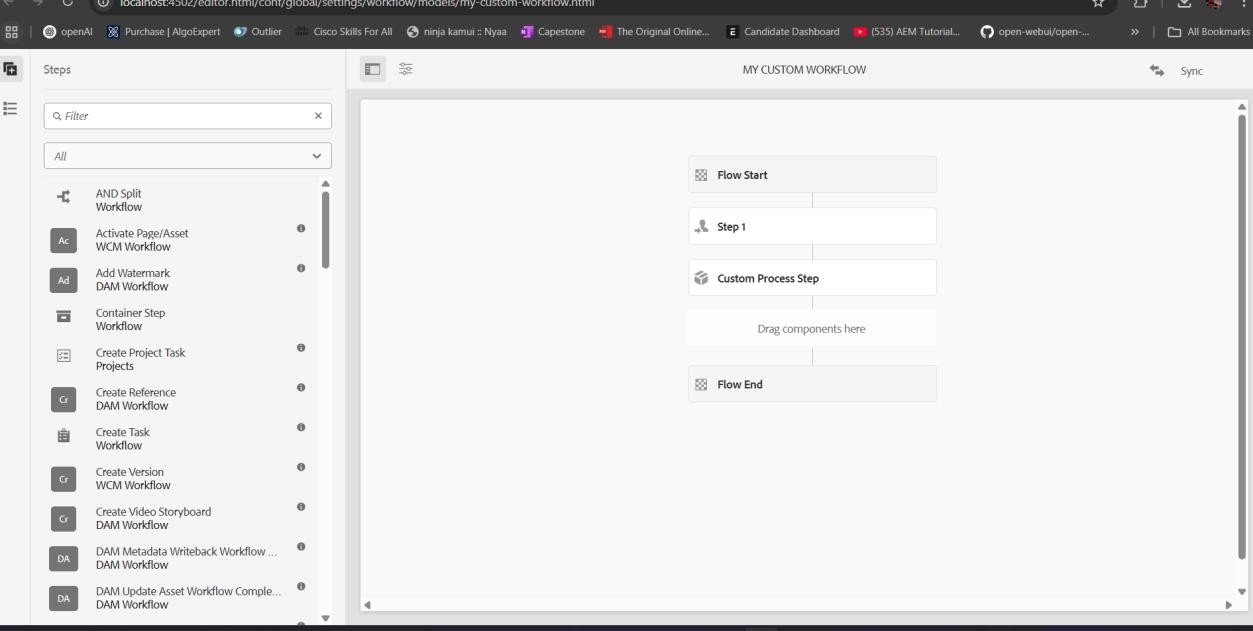
****

**Steps to Check:**

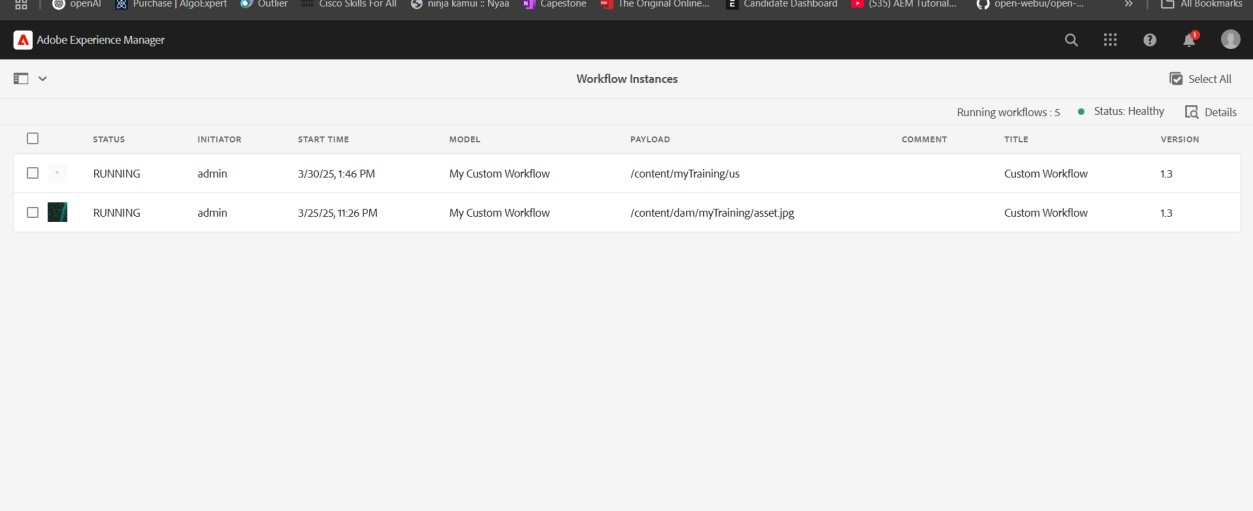
* + Go to **AEM Start → Tools → Workflow → Models**.
  + Check if the **"my custom workflow"** is listed.
  + Open it and ensure the **custom workflow process step** is configured properly.
  + Trigger the workflow on a page by:
    - Navigating to **Sites → Select a Page**.
    - Click on **Page Properties → Start Workflow** and choose your custom workflow.
  + **Check Logs**: o Open AEM **error.log** (crx-quickstart/logs/error.log).
    - Look for log statements that print the **page title**.

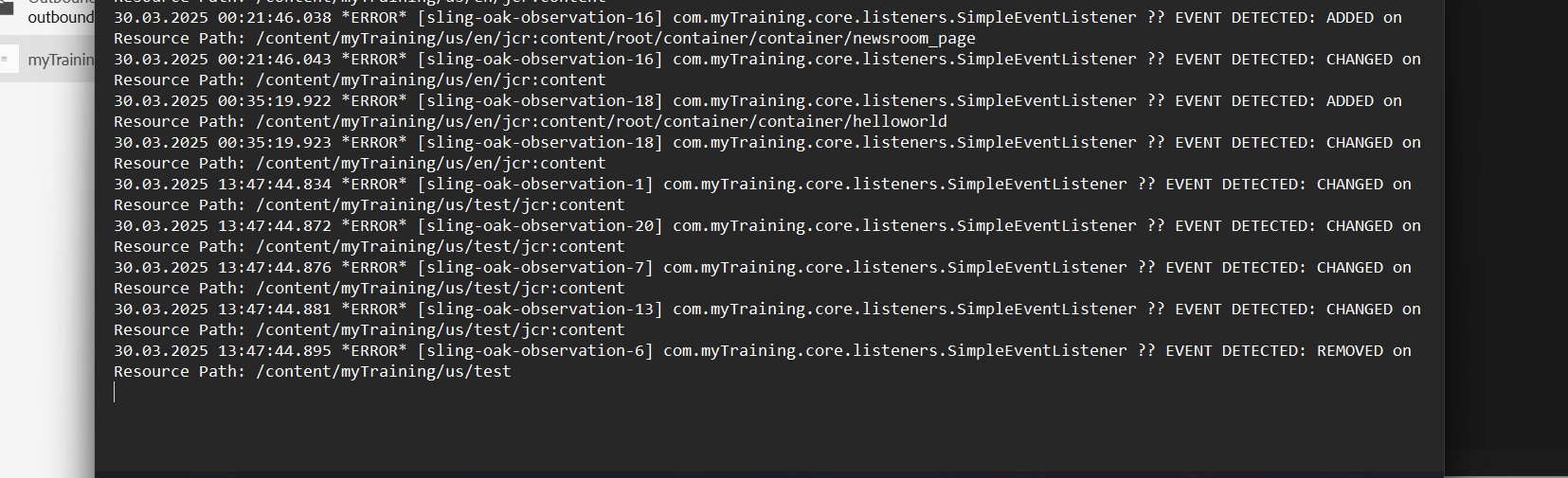




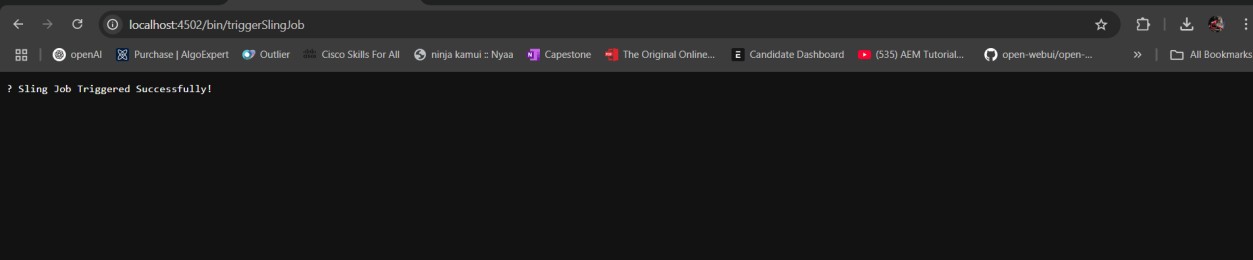


1. **Verify Event Handler Execution Steps to Check:** 
   * Perform an action that triggers the event (e.g., creating, modifying, or deleting a page or an asset).
   * **Check Logs:** o Open error.log and look for log messages related to the resource path.

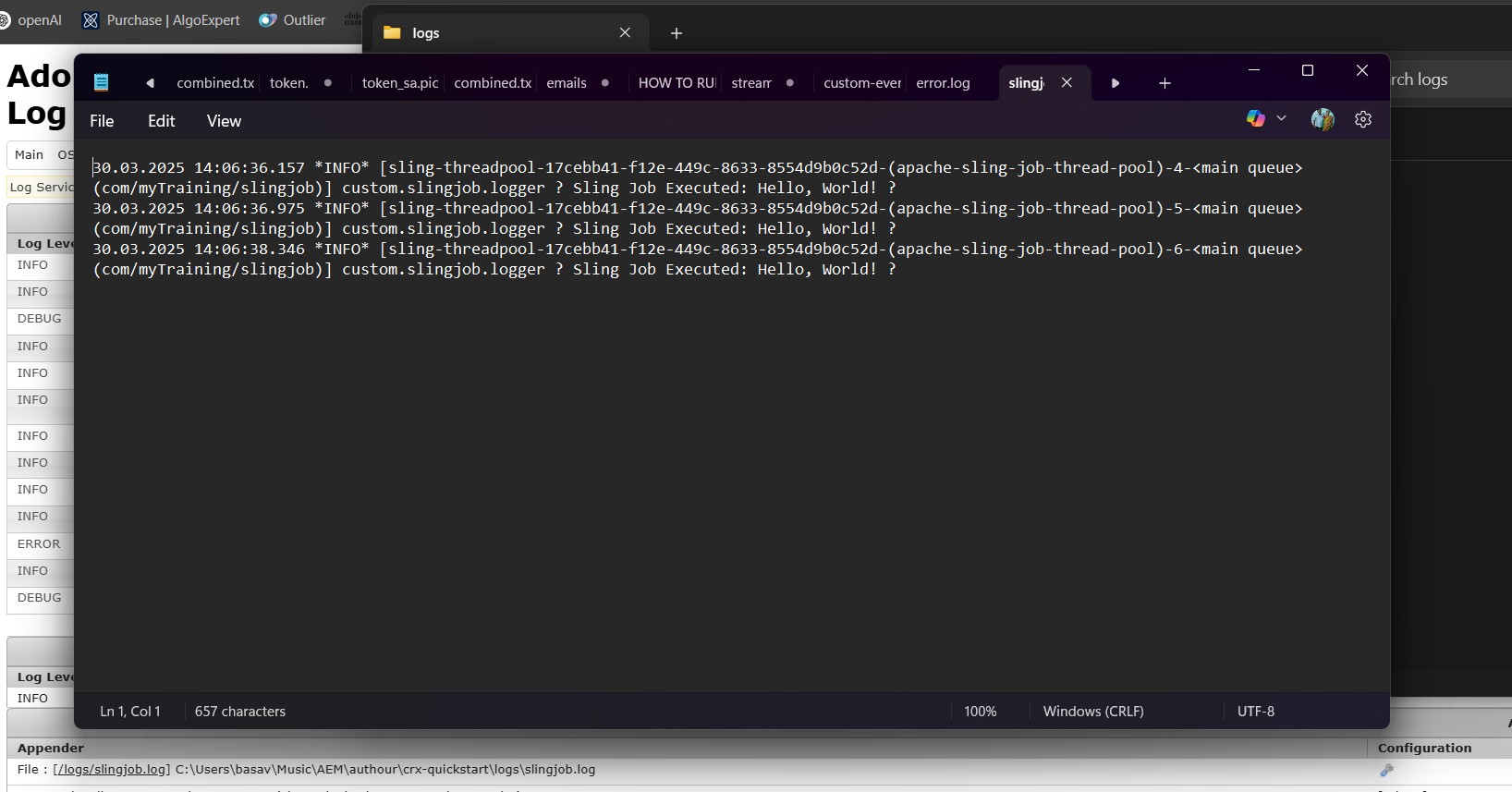




1. **Verify Sling Job Execution Steps to Check:** 
   * Go to **AEM Start → Tools → Operations → Jobs**.
   * Ensure your custom **Sling Job** is listed.
   * **Check Logs:**
     + Open error.log and look for **"Hello World"** message.



Error.log



1. **Verify Scheduler Execution Steps to Check:** 
   * Wait for **5 minutes** after deploying the scheduler.
   * **Check Logs:** o Open error.log and look for **"Yellow World"** printed at every 5-minute interval.

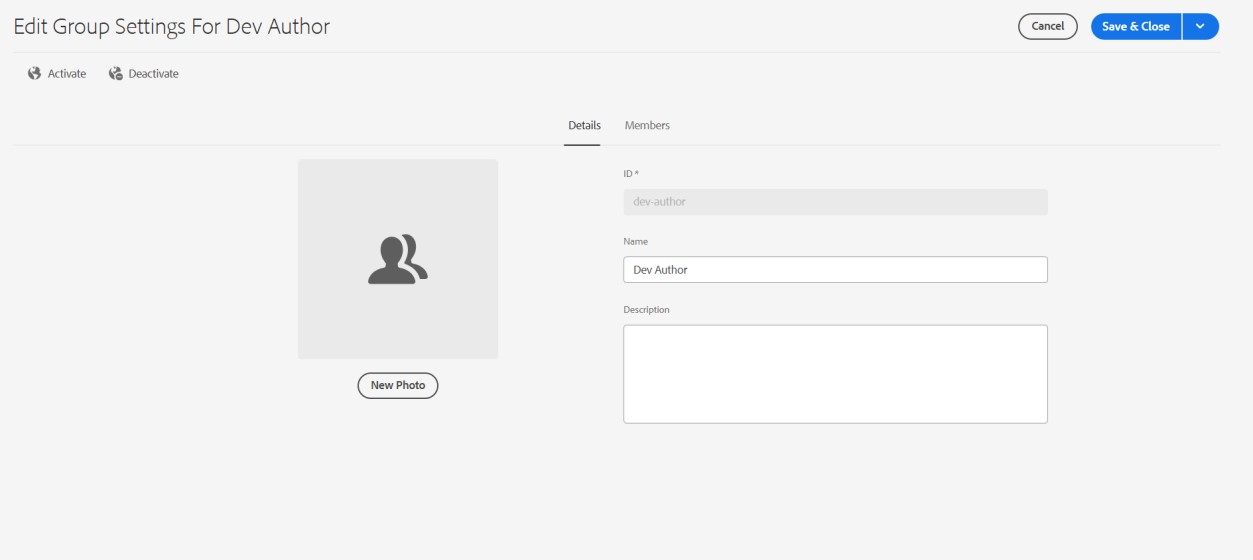
**Expected Output in Logs:**

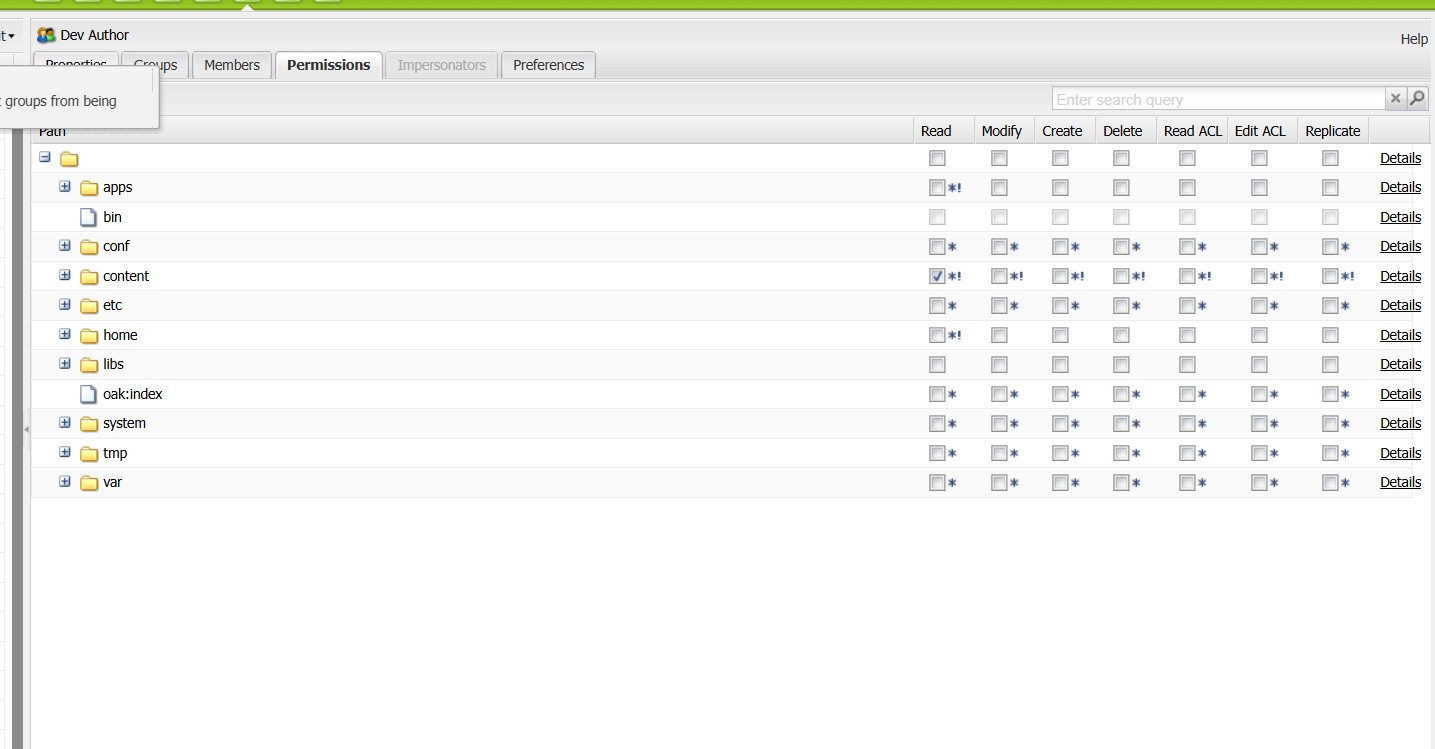
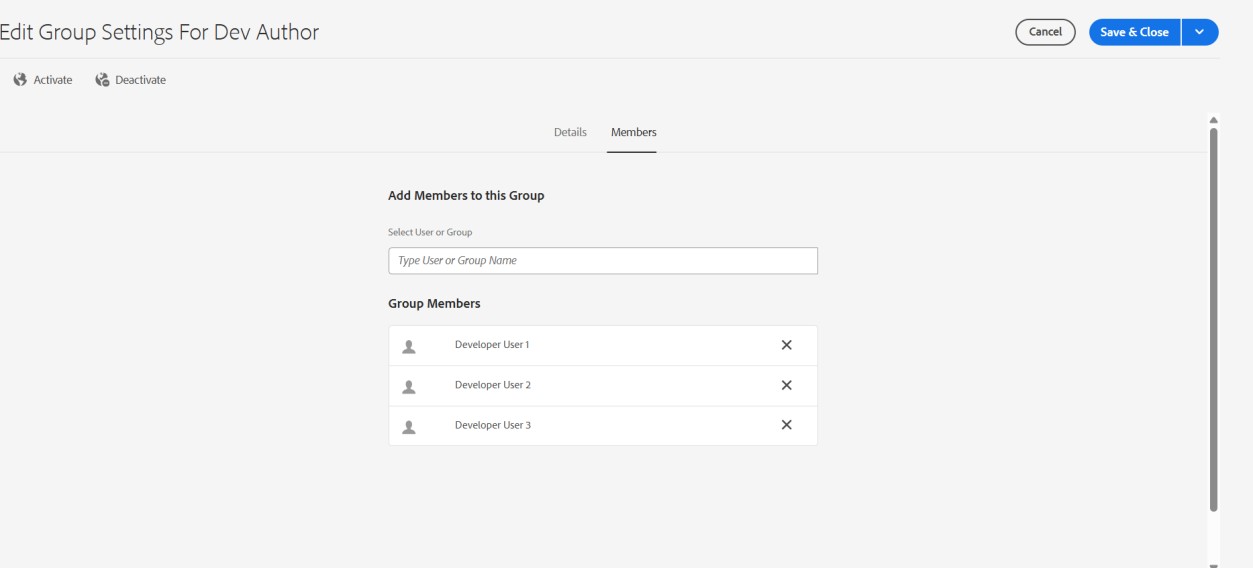
Hello World printed by Scheduler

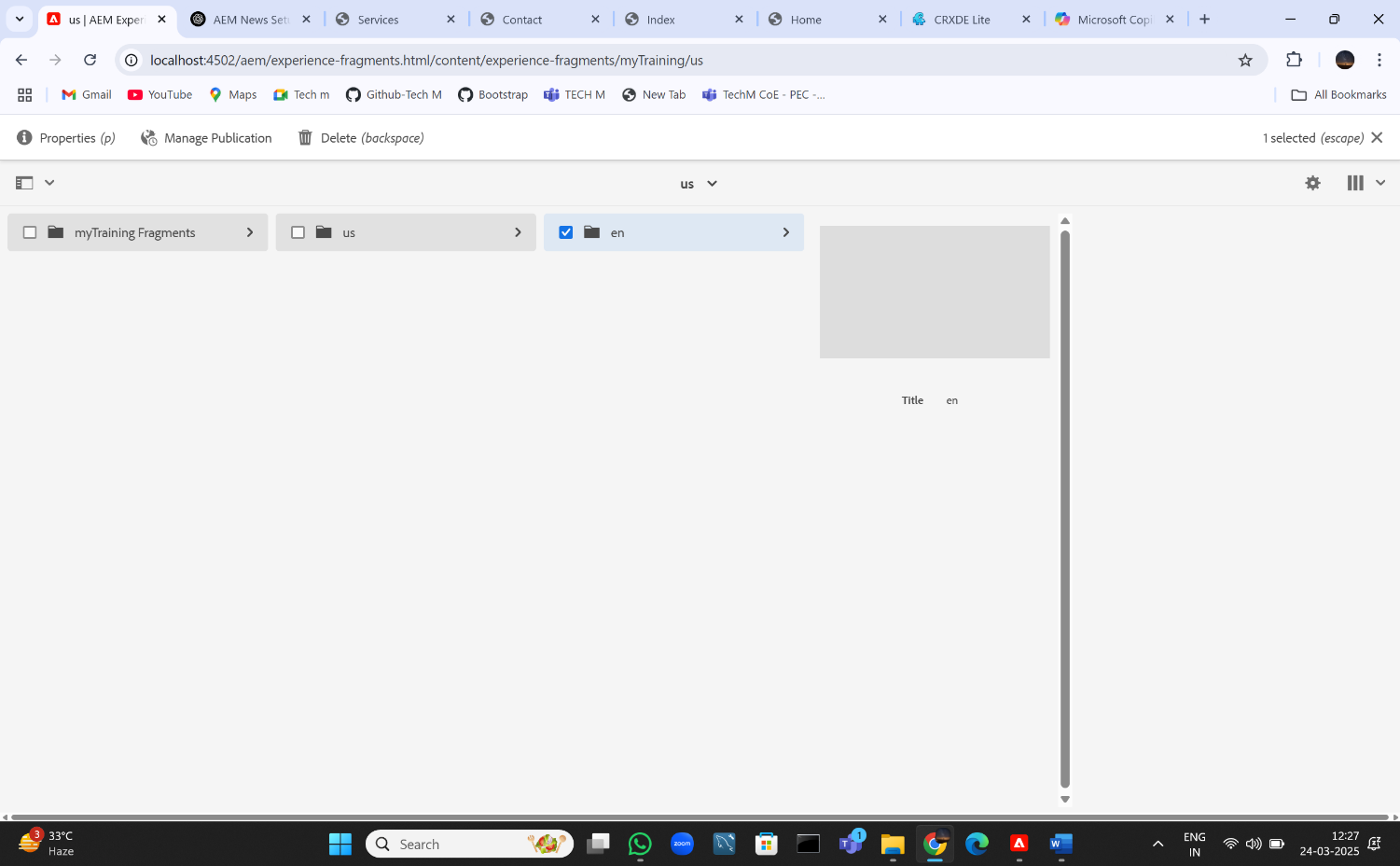
1. **Verify User Group & Permissions Steps to Check:** 
   * Go to **AEM Start → Tools → Security → Groups**.
   * Find **"Dev Author"** group and open it.
   * Ensure the three users are **added**.
   * Check **Permissions:**
     + Go to **/useradmin** (http://localhost:4502/useradmin). o Select the group and verify **Read access** is set for /content and /dam.
     + Ensure **Replication Access** is enabled under **Replicator Permissions**.

**Test Permissions by Logging in as a User in the Group**

* + Login with one of the created users.
  + Try to **view** a page under /content and an asset in /dam.
  + Try **editing** or **deleting** content (should be restricted).
  + Try **replicating** content (should be allowed).





****

**Date:** 26/03/2025

**MSM (Multi-Site Manager):** Facilitates managing multiple websites using shared content or structure.  
**Blueprint:** A template used to create websites with predefined structure and content.  
**Live Copy:** A replicated version of the blueprint inheriting changes.  
**Rollout:** Synchronizes updates from blueprint to live copies.  
**Language Copy:** Site adaptation for different languages.