Date : 19.3.25

1. What is DAM and Why Do We Use It?

- **DAM (Digital Asset Management)** is a system used to organize, store, and manage digital assets like images, videos, documents, and other media files.
- **Centralized Storage:** Keeps all digital assets in one place for easy access.
- **Efficient Management:** Helps organize assets with metadata, tags, and categories.
- **Version Control:** Tracks changes and maintains different versions of assets.
- **Improved Collaboration:** Enables teams to share and access assets seamlessly.
- **Enhanced Workflow:** Integrates with content management systems like AEM for smooth content delivery.

2. What Are Renditions?

Retentions refer to policies that manage the lifecycle of content, specifying how long items are kept before they are archived or deleted.

- For **content versioning** and **workflow management**.
- Ensures compliance with data retention regulations.
- Helps in automatic cleanup of old or unused data.

In DAM (Digital Asset Management):

Retentions control how long digital assets (like images, videos) are kept in the repository before they are archived or purged.

- Manages asset lifecycle, including versions and metadata.
- Automates the deletion of outdated or unnecessary assets.
- Ensures efficient storage management.

3. Add FirstName and LastName Fields in HelloWorld Component

1.Open the HelloWorld Component Dialog in AEM Component Editor:

- Navigate to the AEM Author environment.

- Open the page where the HelloWorld component is placed.

- Click on the component to open its dialog. This dialog allows you to input data for the component.

2. **Add Two Fields in the Dialog:**

- In the dialog configuration (usually in CRX/DE), add the following fields:

  o **First Name (firstName):** A text field where users can enter their first name.

  o **Last Name (lastName):** Another text field for the last name.

- These fields will be represented as form inputs in the dialog, allowing content authors to provide values.
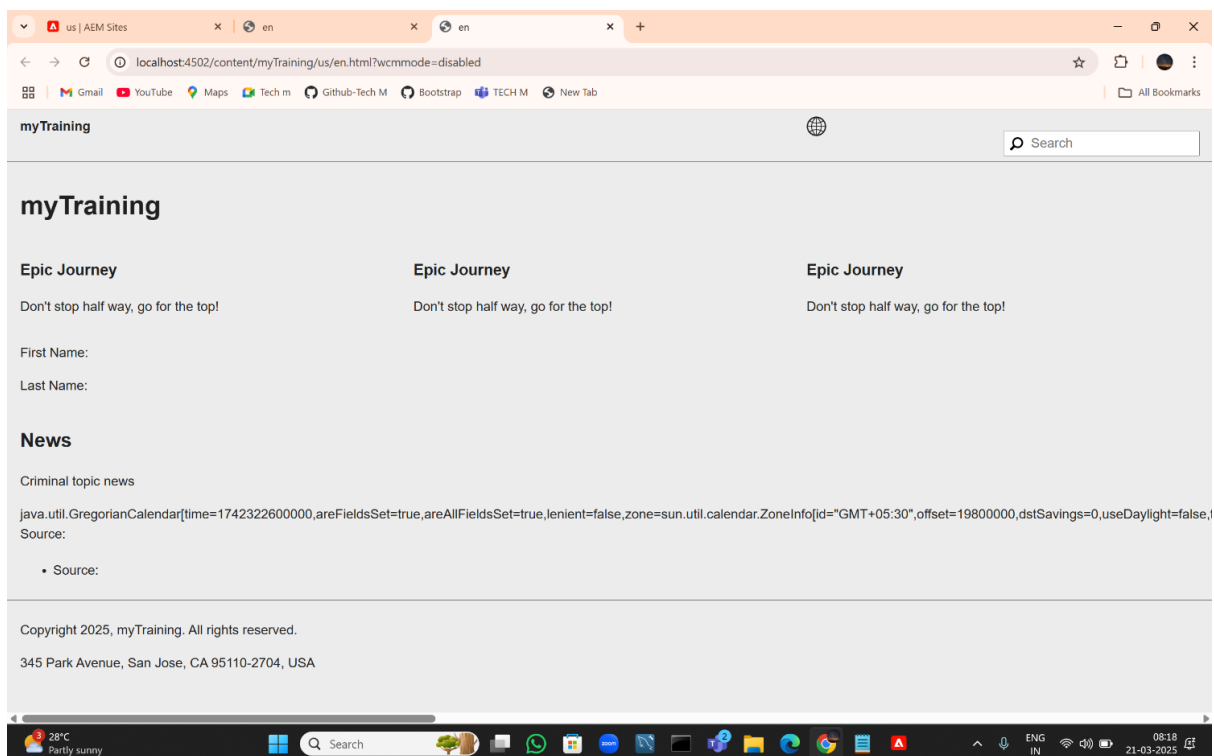
3. **Modify helloworld.html to Display These Values:**

- Open the helloworld.html file in the component folder.

- Update the code to print the entered values using **HTL (Sling Template Language)** and the properties object.

4. **Example Code:**

```
<p>First Name: ${properties.firstName}</p>
```

```
<p>Last Name: ${properties.lastName}</p>
```


5. **Use @ValueMapValue in HelloWorldModel**

```java
package com.example.core.models;

import org.apache.sling.models.annotations.DefaultInjectionStrategy;

import org.apache.sling.models.annotations.Inject;

import org.apache.sling.models.annotations.ValueMapValue;

import org.apache.sling.models.annotations.Model;

@Model(adaptables = Resource.class, defaultInjectionStrategy =
DefaultInjectionStrategy.OPTIONAL)

public class HelloWorldModel {
```

```
@ValueMapValue

  private String firstName;

@ValueMapValue

  private String lastName;

public String getFirstName() {

    return firstName;

  }

public String getLastName() {

    return lastName;

  }

}
```

6. Why Use Package Manager? (Creating Packages)

The **Package Manager** in AEM is a powerful tool used for managing, creating, and deploying **content packages**. These packages help in transferring content, code, and configurations between different AEM environments (like from development to production).

- **Content Migration:** Easily move content, components, templates, and assets from one AEM instance to another.
- **Code Deployment:** Package Java classes, OSGi configurations, and client libraries for deployment across environments.
- **Version Control:** Manage different versions of content and code to ensure consistency during updates.
- **Simplified Backups:** Create backup packages of your AEM environment for recovery or replication purposes.
- **Automation-Friendly:** Integrate with CI/CD pipelines to automate deployments in large-scale projects.

    How to Create a Package:

    - Go to **AEM Package Manager**:
    - Click on **"Create" → "Package"**.
    - Add the required components, content, or code.
    - Build the package and deploy it to the target environment.

7. Configure Replication Agent and Publish Page

1. Configure the Replication Agent:

- **Go to AEM Web Console:**

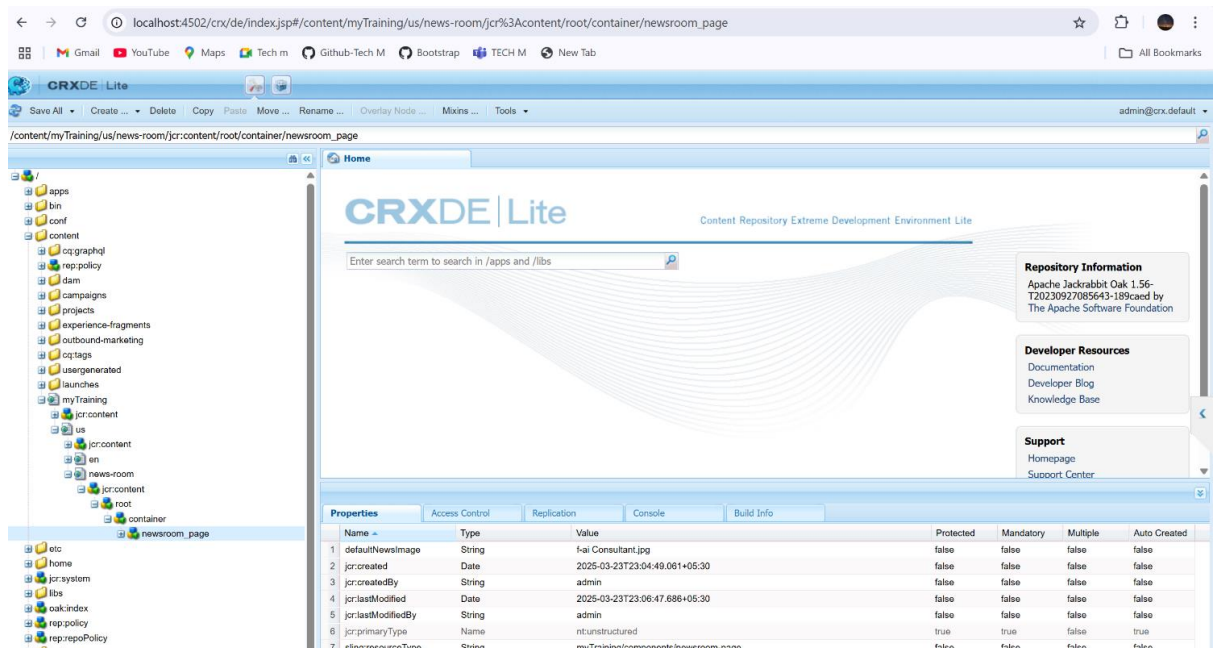http://<aem-host>:<port>/system/console/agentManager.html

- Create a New Agent:
- Click on **"Create"** to add a new replication agent.
- Fill in the details:

- **Name:** e.g., publish-agent

- **Enabled:** Check the box to enable the agent

- **Replication URL:** Provide the URL of the Publish instance (e.g., http://<publish-host>:<port>/bin/receive)

- Click **"Save"** to create the agent.

2. Publish the Page:

- **Go to AEM Author Environment:**

  o Open the page you want to publish.

- **Publish the Page:**

  o Click on the **"Page"** menu in the top toolbar.

  o Select **"Publish"** from the dropdown.

  o Confirm the publishing action.

3. Verify the Publish:

- Go to the **Publish Environment**:

  http://<publish-host>:<port>/content/<your-site>

- Check if the page is live and accessible.

myTraining

Epic Journey

Don't stop half way, go for the top!

Epic Journey

Don't stop half way, go for the top!

Epic Journey

Don't stop half way, go for the top!

First Name:

Last Name:

News

Criminal topic news

java.util.GregorianCalendar[time=1742322600000,areFieldsSet=true,areAllFieldsSet=true,lenient=false,zone=sun.util.calendar.ZoneInfo[id="GMT+05:30",offset=19800000,dstSavings=0,useDaylight=false,
Source:

- Source:

345 Park Avenue, San Jose, CA 95110-2704, USA