

AEM ASSIGNMENT DAY-3

1. Create a New Component with News Title, News Detail, Published Date, and Source Using Sling Model

Answer:

We will create a News Component using the Sling Model to display:

- News Title (h2)
- News Detail (p)
- Published Date
- Source

Steps:

1. Create a new component at: /apps/myTraining/components/news
2. Implement a Sling Model for this component.
3. Map component fields: title, detail, date, source.
4. Update news.html to display these fields.

Code Snippets:

NewsModel.java

```
@Model(adaptables = Resource.class)
public class NewsModel {
    @Inject
    private String title;

    @Inject
    private String detail;

    @Inject
    private String date;

    @Inject
    private String source;

    public String getTitle() { return title; }
    public String getDetail() { return detail; }
    public String getDate() { return date; }
    public String getSource() { return source; }
}
```

news.html

```
<div class="news-item">
    <h2>${newsModel.title}</h2>
    <p>${newsModel.detail}</p>
    <span>${newsModel.date}</span>
    <small>${newsModel.source}</small>
</div>
```

SCREENSHOTS

The screenshot shows the CRXDE Lite interface for the news.html component. The left sidebar lists various AEM components like button, carousel, container, etc. The main area shows the XML code for the news.html component:

```

<div class="news-component" data-sly-use.newsModel="com.myTraining.core.models.News">
    <div class="cmp-news-title">
        <h1>${newsModel.newsHeadline}</h1>
    </div>
    <div>
        <img alt="${newsModel.newsImg}" alt="${newsModel.title}" width="auto" height="400">
    </div>
    <div class="cmp-news-article-data1">
        <p>News Published on: ${dd MMM, YYYY} @ format=newsModel.newsPublishDate,type='datetime'</p>
        <p>${newsModel.newsDetail @ context='html'}</p>
    </div>
    <div data-sly-list.item="${newsModel.newsTitle}">
        <p>${item}</p>
    </div>

```

The Properties tab shows the following configuration:

Name	Type	Value	Protected	Mandatory	Multiple
1 jcr:created	Date	2025-03-24T16:14:29.006+05:30	true	false	false
2 jcr:createdBy	String	admin	true	false	false
3 jcr:primaryType	Name	nt:file	true	true	false

The screenshot shows the AEM authoring interface with the news component being edited. The Content Tree on the left shows the news component under the myTraining container. The right panel displays the News dialog with the following fields:

- News Headline:** EARTH QUAKE
- News Detail Article:** Earthquakes can strike suddenly and without warning. An earthquake is a violent and abrupt shaking of the ground, caused by movement between tectonic plates along a fault line in the earth's crust. Earthquakes can result in ground shaking, soil liquefaction, landslides, fissures, avalanches, fires and tsunamis.
- News Image:** A thumbnail image of a road cracked by an earthquake.
- News Publish Date:** 01 April, 2025
- News Titles:** SOON, goww

The preview pane shows a visual representation of the news item with the headline "EARTH QUAKE" and the image of the cracked road.

2. Create a Multi-field Component (Multiple News) Using Sling Model

Answer:

A multi-field component allows authors to dynamically add multiple news items.

Steps:

1. Create a multi-field component at: /apps/myTraining/components/multi-news
2. Implement a Sling Model List.
3. Map fields: title and source.
4. Update multi-news.html to loop through news items.

Code Snippets:

MultiNewsModel.java

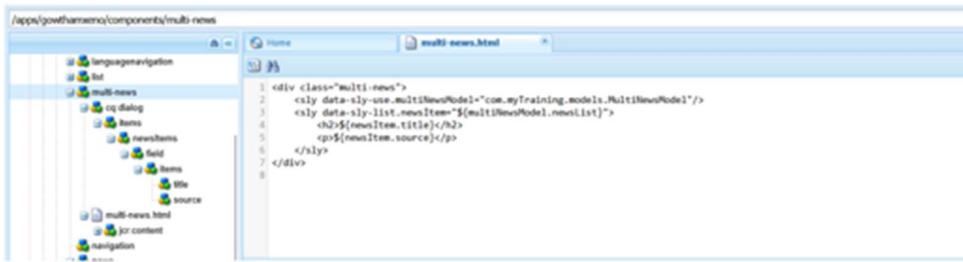
```
@Model(adaptables = Resource.class)
public class MultiNewsModel {
    @Inject
    @Children
    private List<NewsModel> newsList;

    public List<NewsModel> getNewsList() { return newsList; }
}
```

multi-news.html

```
<div class="multi-news">
    <sly data-sly-list.newsItem="${multiNewsModel.newsList}">
        <h2>${newsItem.title}</h2>
        <p>${newsItem.source}</p>
    </sly>
</div>
```

1. Multi-news



3. Create Clientlibs for News Component

Answer:

Clientlibs are used to apply styles and scripts to components.

Steps:

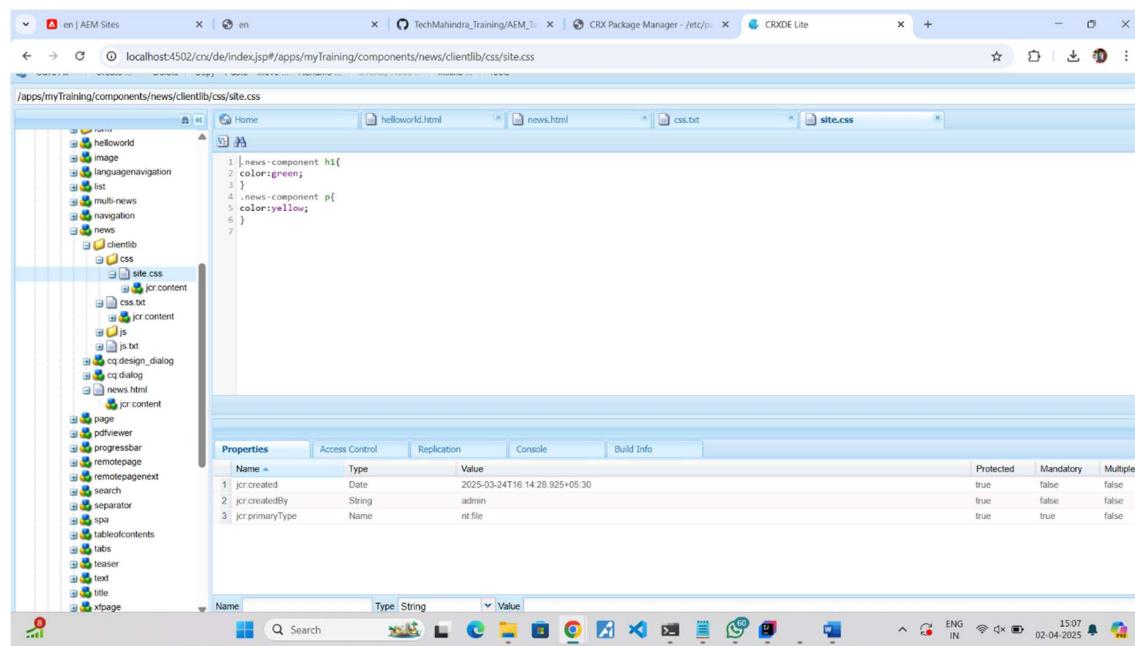
1. Create a clientlib at: /apps/myTraining/clientlibs/news
2. Include CSS and JS files.
3. Add clientlib categories in the component.

Code Snippet:

|clientlibs/news/css.txt

news.css

```
.news-item h2 { color: green; }
.news-item p { color: yellow; }
.news-item span { color: black; }
```



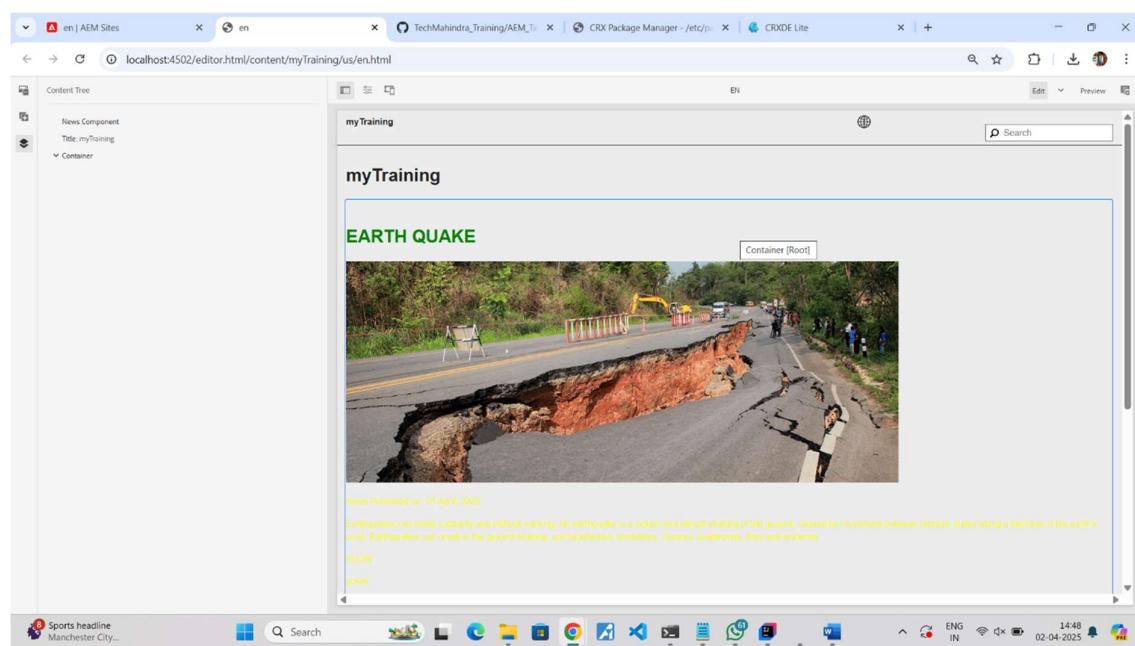
4. Apply Green Color to Heading (h2), Yellow to News Detail (p), and Black to Date (span)

Answer:

This is done using CSS inside clientlibs/news.

Code Snippet:

```
.news-item h2 { color: green; }
.news-item p { color: yellow; }
.news-item span { color: black; }
```



5. Add Component Style (cq-news-component) in News Component

Answer:

We will add a custom style name to the News Component.

Steps:

1. Navigate to: /apps/myTraining/components/news in CRXDE Lite.
2. Update cq:dialog to include the style property.
3. Modify news.html to include the CSS class.

Code Snippet:

news.html

```
<div class="cop-news-component">
    <h2>${properties.newsTitle}</h2>
    <p>${properties.newsDescription}</p>
</div>
```

The screenshot shows the AEM authoring interface with the following details:

- Left Sidebar:** Shows the site navigation tree under "content". The "news.html" component is selected.
- Central Area:** The code editor displays the content of the "news.html" file:

```
13     <p class="news-date">News Published on: ${'dd MMMM, YYYY' @ format=newsModel.newsPublishDate,type='datetim}
14     <p> ${newsModel.newsDetail @ context='html'}</p>
15   </div>
16
17   <div data-sly-list.item="${newsModel.newsTitle}">
18     <p>${item}</p>
19   </div>
20   <div class="cop-news-component">
21     <h2>News Title : ${properties.newsTitle}</h2>
22     <p>News Description : ${properties.newsDescription}</p>
23   </div>
24
25 </div>
26 </div>
```
- Properties Panel:** Shows the component's properties:

Name	Type	Value	Protected
1 sling:resourceType	String	granite/ui/components/coral/foundation/form/textarea	false
2 name	String	/newsDescription	false

6. Create a Base Page Component and Add Metadata for OG Tags

Answer:

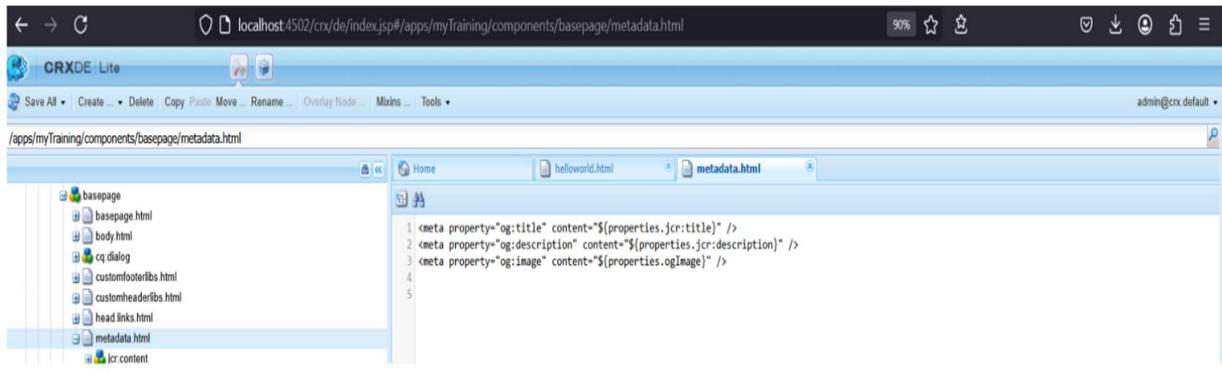
We will create a metadata.html file to print OG meta tags.

Steps:

1. Create metadata.html inside the base page component.
2. Add OG metadata properties.
3. Link metadata.html in basepage.html.

Code Snippet (metadata.html)

```
<meta property="og:title" content="${pageProperties.ogTitle}">
<meta property="og:description" content="${pageProperties.ogDes<
<meta property="og:image" content="${pageProperties.ogImage}">
```



7. Create Custom Page Properties (Global Properties)

Answer:

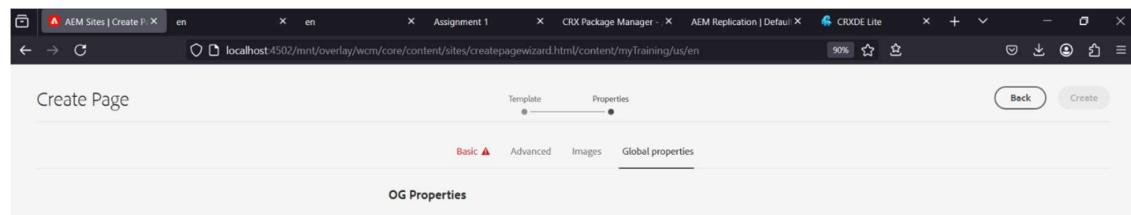
We will add three custom fields (`og:title`, `og:description`, `og:image path`) in Page Properties.

Steps:

1. Navigate to: /apps/myTraining/components/page/basepage/cq:dialog.
2. Add a tab named Global Properties.
3. Create fields:
 - o `ogTitle`
 - o `ogDescription`
 - o `ogImage`

SCREENSHOTS

1.GLOBAL PROPERTIES



8. What is extraClientLibs and How to Use It in a Multi-field Component?

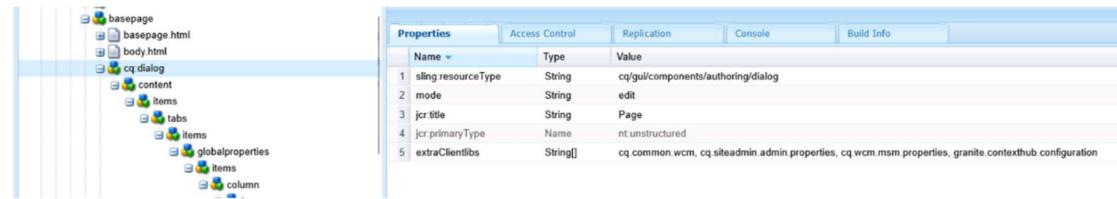
Answer:

`extraClientLibs` is used to include additional CSS & JavaScript files in a component.

Code Snippet:

```
<extraClientLibs jcr:primaryType="nt:unstructured">
    <css>
        <item0>clientlibs/myTraining/styles.css</item0>
    </css>
</extraClientLibs>
```

1. extraClientLibs



The screenshot shows the AEM authoring interface. On the left is a tree view of page components under 'cq dialog'. On the right is a properties dialog for the selected node. The 'Properties' tab is selected, showing the following table:

Name	Type	Value
1 sling resourceType	String	cq/gui/components/authoring/dialog
2 mode	String	edit
3 jcr title	String	Page
4 jcr primaryType	Name	nt:unstructured
5 extraClientlibs	String[]	cq:common.wcm, cq:siteadmin.admin.properties, cq:wcm.msm.properties, granite.contexthub.configuration

Conclusion

This assignment covered essential AEM development tasks, including: 

Creating a News Component with Sling Models

-  [Implementing a Multi-field Component](#)
-  [Using Clientlibs for Styling & Scripts](#)
-  [Adding Custom Styles to Components](#)
-  [Creating a Base Page with OG Metadata](#)
-  [Defining Global Page Properties](#)
-  [Using extraClientLibs for Additional Resources](#)