

# AEM Training - Day 3 Assignment (20-03-2025)

---

## 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**

java

CopyEdit

```
@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

html

CopyEdit

```

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

```

---

## 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

java

CopyEdit

```
@Model(adaptables = Resource.class)

public class MultiNewsModel {

    @Inject

    @Children

    private List<NewsModel> newsList;


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

}
```

### multi-news.html

html

CopyEdit

```
<div class="multi-news">

    <sly data-sly-list.newsItem="${multiNewsModel.newsList}">

        <h2>${newsItem.title}</h2>

        <p>${newsItem.source}</p>

    </sly>

</div>
```

---

### 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 Snippets

**clientlibs/news/css.txt**

CopyEdit

news.css

**news.css**

css

CopyEdit

```
.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:**

css

CopyEdit

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

---

## 5. Add Component Style (cop-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

html

CopyEdit

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

---

## 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 **basepage component**.
2. Add **OG metadata properties**.

3. Link metadata.html in **basepage.html**.

#### **Code Snippet (metadata.html)**

html

CopyEdit

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

### **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:**

xml

CopyEdit

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

---

## 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