

**20-03-2025**

## **1. News Component (Sling Model)**

**Fields:**

- News Title → display as <h2>
- News Detail → display as <p>
- Published Date
- Source

**news.html**

html

CopyEdit

```
<div class="cop-news-component">
  <h2 class="news-title">${newsModel.title}</h2>
  <p class="news-detail">${newsModel.detail}</p>
  <div class="news-date">Published on: ${newsModel.publishedDate}</div>
  <div class="news-source">Source: ${newsModel.source}</div>
</div>
```

**NewsModel.java**

java

CopyEdit

```
@Model(adaptables = Resource.class, defaultInjectionStrategy = DefaultInjectionStrategy.OPTIONAL)
public class NewsModel {
```

```
    @Inject
```

```
    private String title;
```

```
    @Inject
```

```
    private String detail;
```

```
    @Inject
```

```
    private String publishedDate;
```

```

@Inject
private String source;

public String getTitle() { return title; }
public String getDetail() { return detail; }
public String getPublishedDate() { return publishedDate; }
public String getSource() { return source; }
}

```

---

## 2. Multifield Component (Multiple News Items)

### newslst.html

html

CopyEdit

```

<div class="news-list">
  <ul data-sly-list="${newsListModel.newsList}">
    <li>
      <strong>${item.title}</strong> - <em>${item.source}</em>
    </li>
  </ul>
</div>

```

### NewsListModel.java

java

CopyEdit

```

@Model(adaptables = Resource.class)
public class NewsListModel {

    @Inject
    @Named("news")
    private List<NewsItem> newsList;

    @Model(adaptables = Resource.class)

```

```

public static class NewsItem {

    @Inject private String title;

    @Inject private String source;


    public String getTitle() { return title; }

    public String getSource() { return source; }

}


public List<NewsItem> getNewsList() {

    return newsList;

}

}

```

---

### 3. Create Clientlibs

**Path:** /apps/your-project/components/news/clientlibs

**Add cq:ClientLibraryFolder = true**

**Add categories property:**

txt

CopyEdit

your.project.news

**Add CSS file (news.css)**

css

CopyEdit

```
.cop-news-component h2 {
```

```
    color: green;
```

```
}
```

```
.cop-news-component p {
```

```
    color: yellow;
```

```
}
```

```
.cop-news-component .news-date {
```

```
    color: black;
```

```
}
```

**news.html include clientlib**

html

CopyEdit

```
<cq:includeClientLib categories="your.project.news"/>
```

---

#### 4. Apply Colors to Elements

Already handled in the CSS above.

---

#### 5. Add Component Style (cop-news-component)

**In news component node**

Add:

txt

CopyEdit

sling:resourceSuperType = "core/wcm/components/xxx" (if needed)

cq:styleDefaultClasses = "cop-news-component"

OR use Policy in Editable Template:

1. Go to template → Select Policy → Styles
  2. Add Style Group: News Styles
  3. Add Style: COP News Component → CSS Class: cop-news-component
- 

#### 6. Create Base Page Component with Meta Tags

**basepage.html**

html

CopyEdit

```
<head>
```

```
<meta property="og:title" content="{pageProperties['og:title']}" />
```

```
<meta property="og:description" content="{pageProperties['og:description']}" />
```

```
<meta property="og:image" content="{pageProperties['og:image']}" />
```

```
</head>
```

---

## 7. Custom Page Properties: Global Properties

**Dialog XML** in page component (or use cq:dialog)

xml

CopyEdit

```
<tabs>

  <tab jcr:title="Global Properties">

    <items>

      <textfield

        jcr:primaryType="nt:unstructured"

        fieldLabel="OG Title"

        name="/.og:title"

        xtype="textfield"/>

      <textfield

        jcr:primaryType="nt:unstructured"

        fieldLabel="OG Description"

        name="/.og:description"

        xtype="textfield"/>

      <pathfield

        jcr:primaryType="nt:unstructured"

        fieldLabel="OG Image Path"

        name="/.og:image"

        xtype="pathfield"/>

    </items>

  </tab>

</tabs>
```

---

extraClientlibs is a **component policy setting** used to **load extra client libraries** automatically when a component is used.

**How to Add?** In the component policy (template editor):

- Go to your component policy
- Add extraClientlibs property
- Value: [ "your.project.news" ]

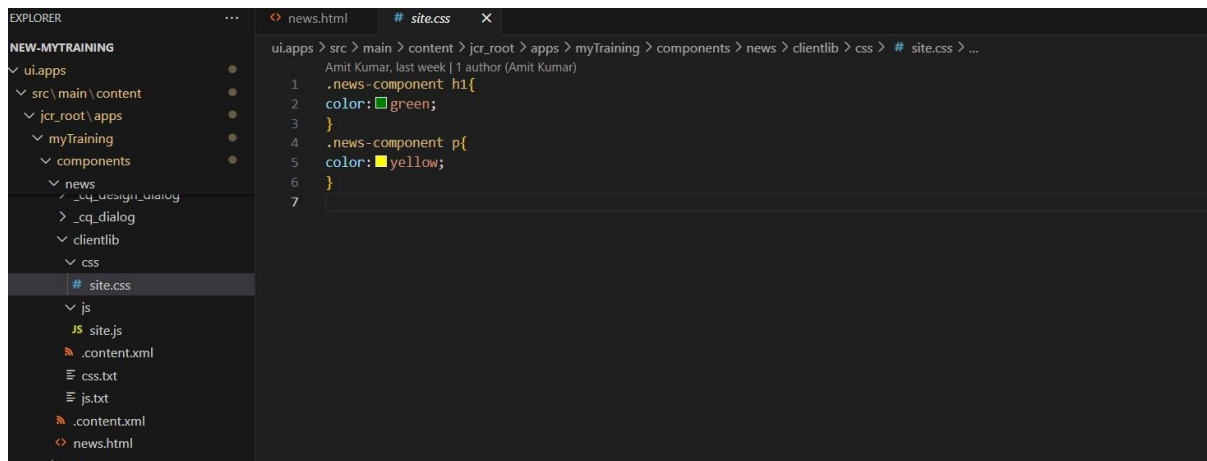
This ensures the clientlib is loaded even without explicitly including via `<cq:includeClientLib>`.

### Apply Styling (CSS)

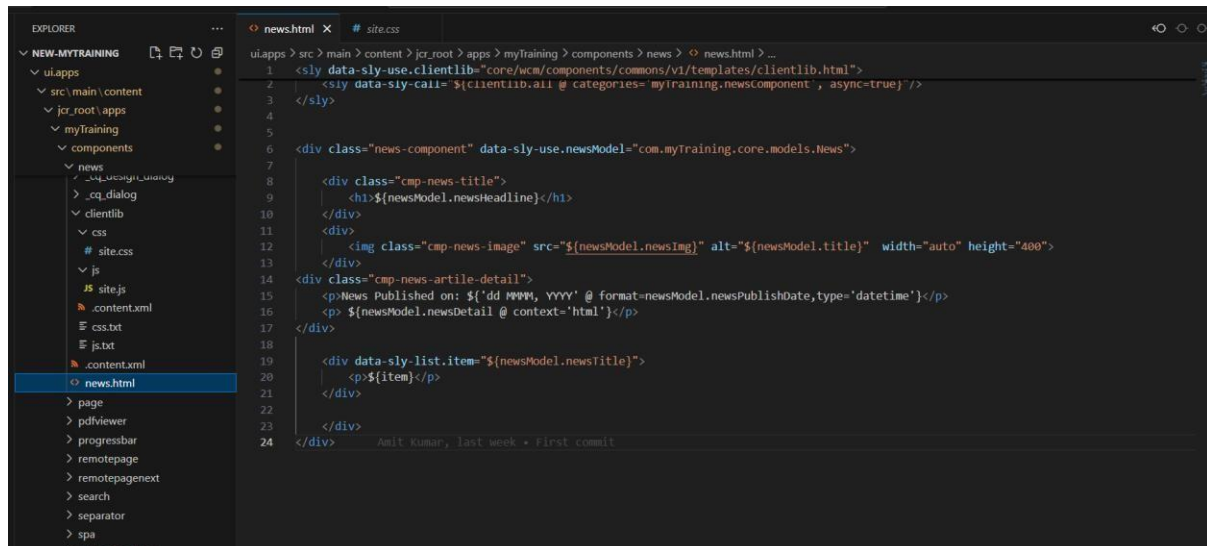
Inspect the rendered page using **Developer Tools (F12) > Elements Tab** and check:

- **Heading (h2) is Green (color: green).**
- **News Detail (<p>) is Yellow (color: yellow).**
- **Date is Black (color: black).**

### Css.txt



### News.html



### Extra ClientLibs (extraClientLibs)

#### How to Check:

- Open the multi-field component's **HTL (HTML Template Language)** file.

```
...  news.html X  # site.css

ui.apps > src > main > content > jcr_root > apps > myTraining > components > news > news.html > div.news-component

1  <sly data-sly-use.clientlib="core/wcm/components/commons/v1/templates/clientlib.html">
2  |   <sly data-sly-call="${clientlib.all @ categories='myTraining.newsComponent', async=true}"/>
3  </sly>
4
5
6  <div class="news-component" data-sly-use.newsModel="com.myTraining.core.models.News">
7
8      <div class="cmp-news-title">
```