

1. Create A new component which have news title(should print as h2), news detail(<p>), published date, source create this component using sling model.

NewsModel.java

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

```
public class NewsModel {
```

```
    @Inject
```

```
    private String text;
```

```
    @Inject
```

```
    private String newsDetail;
```

```
    @Inject
```

```
    private Calendar publishedDate; // FIXED: Changed to Calendar type
```

```
    public String getText() {
```

```
        return text;
```

```
    }
```

```
    public String getNewsDetail() {
```

```
        return newsDetail;
```

```
    }
```

```
    public String getFormattedPublishedDate() {
```

```
        if (publishedDate != null) {
```

```
            SimpleDateFormat sdf = new SimpleDateFormat("dd MMM yyyy");
```

```
            return sdf.format(publishedDate.getTime()); // FIXED: Now works with
```

```
Calendar
```

```
        }
```

```
        return "N/A";
```

```
    }
```

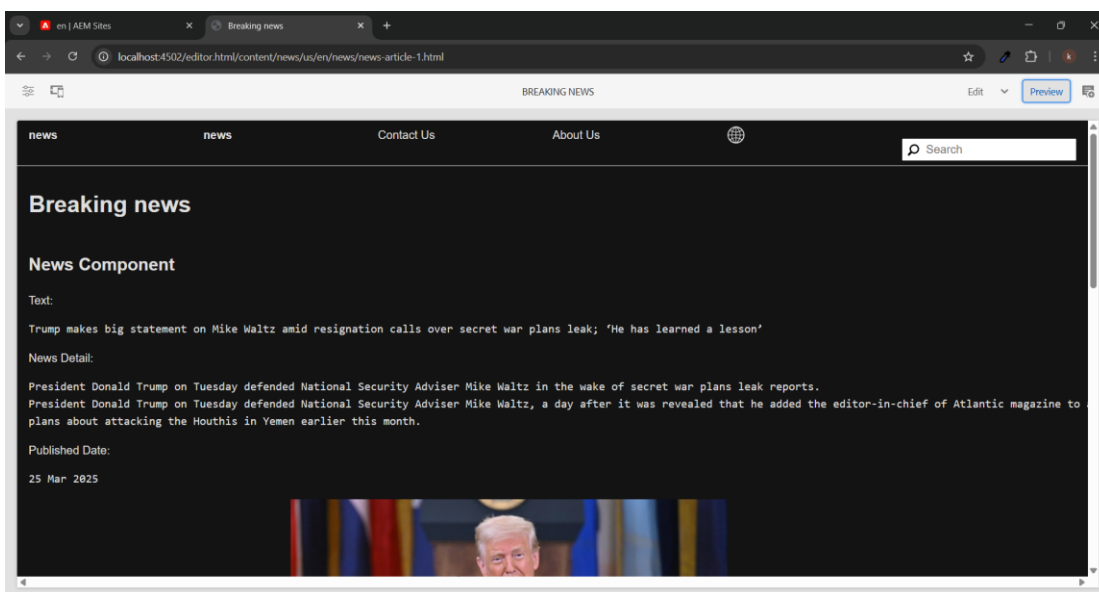
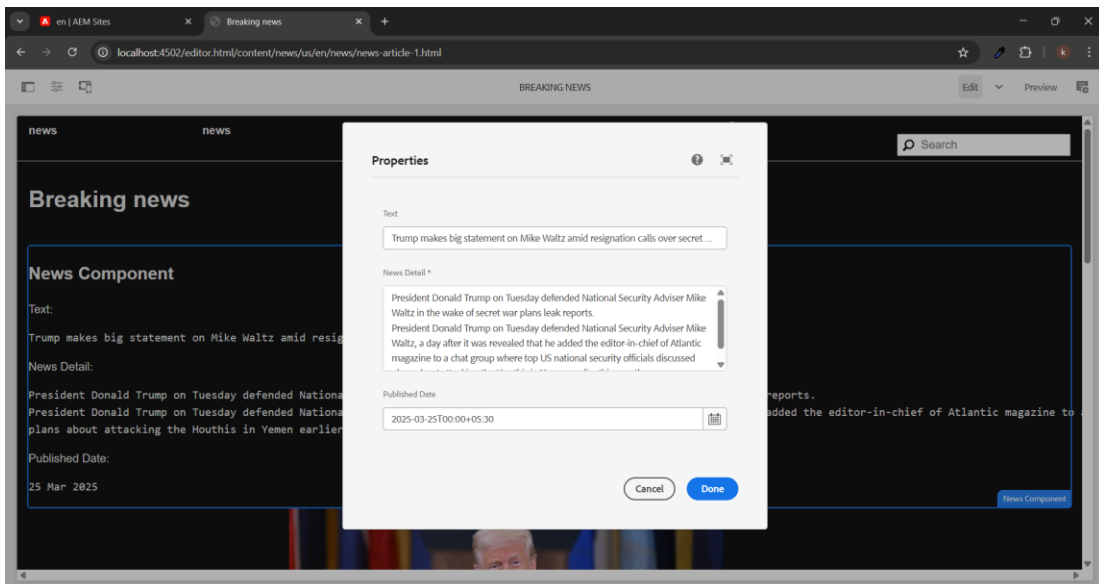
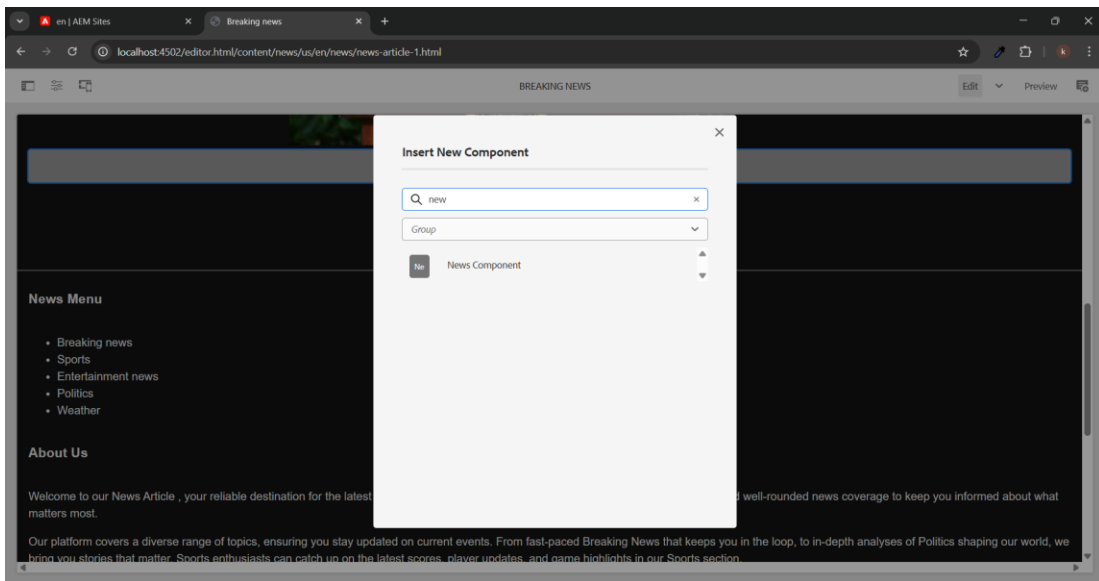
```
}
```

News.html

```
<h2 class="cmp-news__title">News Component</h2>
```

```
<p class="cmp-news__label">Text:</p>
```

```
<p class="cmp-news__label">Published Date:</p>
```



2.Create multi field component(multiple news) using sling model fields: title of new and source of news

MultiField News

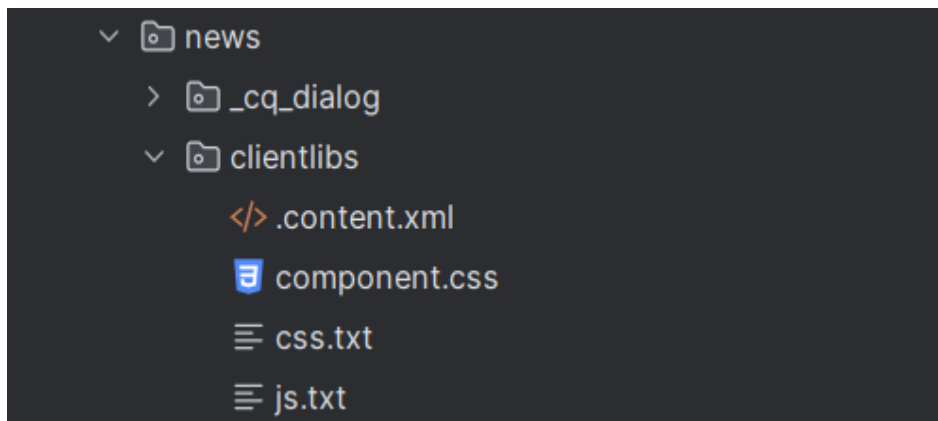
News Items

Title: News1  
Source: Source1

Title: News2  
Source: Source2

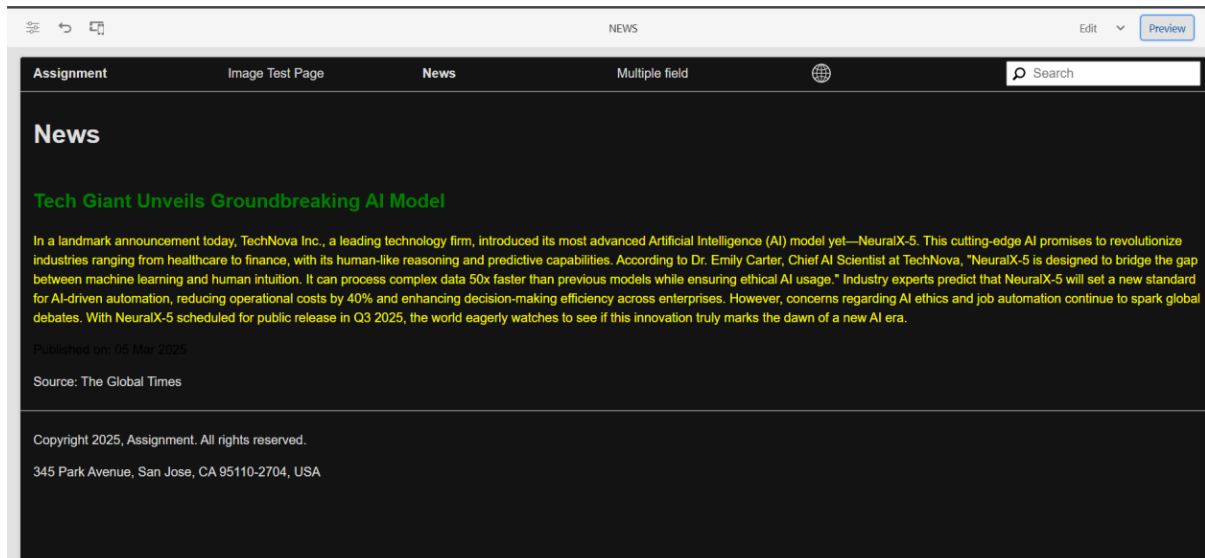
Add

3.Create clientlibs for news component



```
.cmp-news__title {  
    color: green;  
}  
  
.cmp-news__label {  
    color: yellow;  
}  
  
.cmp-news__output {  
    color: black;  
}
```

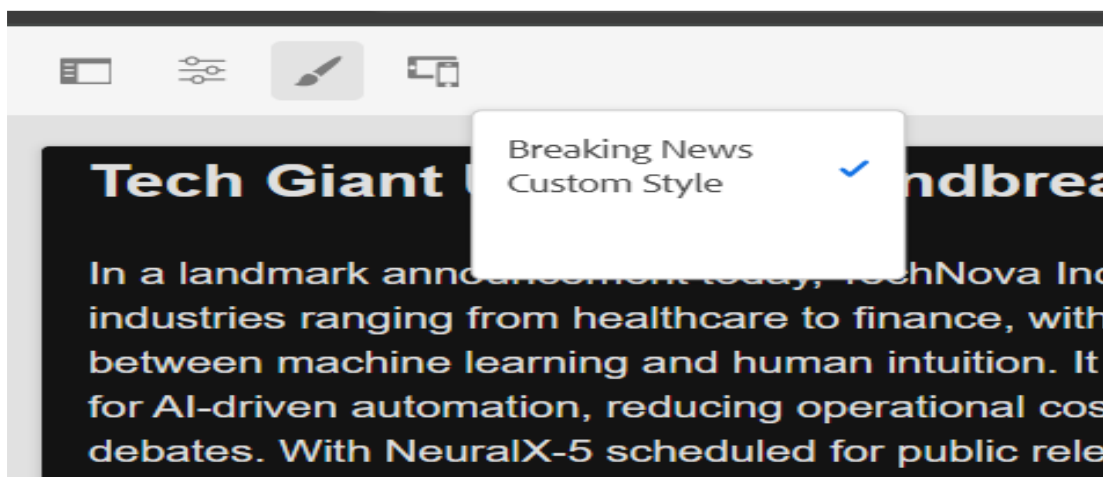
4. Apply Green colour to the heading(h2) and yellow colour to news detail(<p) and date should be black in colour.



5. Add Component style to add custom style name (cop-news-component) in news component

**content.xml**

```
<jcr:root xmlns:jcr="http://www.jcp.org/jcr/1.0"
  xmlns:cq="http://www.day.com/jcr/cq/1.0"
  jcr:primaryType="cq:Component"
  jcr:title="News Component"
  componentGroup="Example Components"
  cq:style="cop-news-component"/>
```



Component Path: /apps/news/components/basepage

### **basepage.html**

```
<html>
<head>
  <link rel="stylesheet" href="/etc.clientlibs/example/clientlibs/news.css"
type="text/css">
  <meta property="og:title" content="{properties.ogTitle}" />
  <meta property="og:description" content="{properties.ogDescription}" />
  <meta property="og:image" content="{properties.ogImage}" />
</head>
<body>
  <sly data-sly-resource="{news' @
resourceType='news/components/news'}"></sly>
</body>
</html>
```

7. Create custom page properties saying Global properties and 3 fields should be there as og:title, og :description and og:image path.

### **dialog.xml**

```
<jcr:root xmlns:jcr="http://www.jcp.org/jcr/1.0"
xmlns:sling="http://sling.apache.org/jcr/sling/1.0"
jcr:primaryType="cq:Dialog"
cq:dialogMode="floating">
  <content jcr:primaryType="nt:unstructured">
    <items jcr:primaryType="nt:unstructured">
      <ogTitle
jcr:primaryType="nt:unstructured"
sling:resourceType="granite/ui/components/coral/foundation/form/textfield"
fieldLabel="OG Title"
```

```
name="./ogTitle"/>
<ogDescription
jcr:primaryType="nt:unstructured"
sling:resourceType="granite/ui/components/coral/foundation/form/textarea"
fieldLabel="OG Description"
name="./ogDescription"/>
<ogImage
jcr:primaryType="nt:unstructured"
sling:resourceType="granite/ui/components/coral/foundation/form/textfield
```

8. What is `extraClientLib` add it this multi field component

The `extraClientLib` property adds additional client libraries (CSS, JS) to the multi-field component, allowing custom styling and JavaScript logic.

Why Use `extraClientLibs`?

- Helps extend functionality without modifying core component clientlibs.
- Useful for adding custom JavaScript or styles specific to a component.
- Allows separation of concerns, keeping core and additional scripts/styles modular.
- Makes debugging and maintenance easier by loading external scripts as needed