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

Step 1: Create the Component

- Open CRXDE Lite (http://localhost:4502/crx/de).
- Navigate to /apps/myTraining/components/ and create a new component named newsComponent.
- Inside newsComponent, create a newsComponent.html file and cq:dialog.

Step 2: Create Sling Model

- Create a Java class under com.myTraining.core.models.

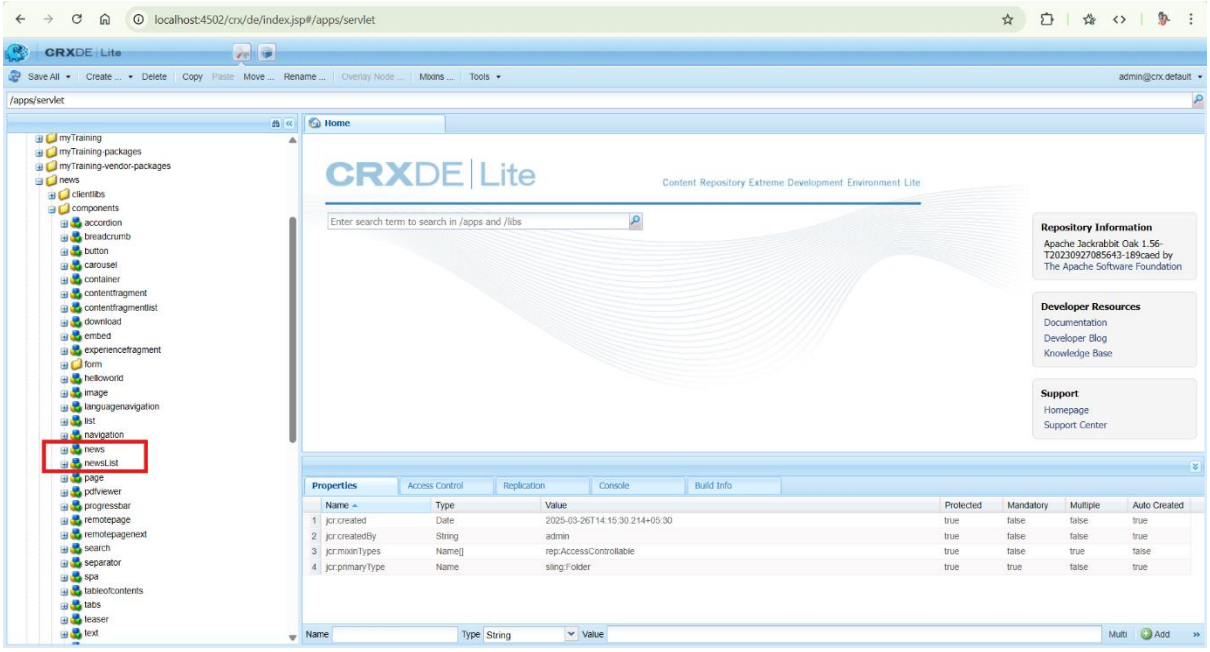
MultiNewsModel.java

```
package com.myTraining.core.models;
import org.apache.sling.api.resource.Resource;
import org.apache.sling.models.annotations.DefaultInjectionStrategy;
import org.apache.sling.models.annotations.Model;
import org.apache.sling.models.annotations.injectorspecific.ChildResource;
import java.util.ArrayList;
import java.util.List;
@Model(adaptables = Resource.class, defaultInjectionStrategy =
DefaultInjectionStrategy.OPTIONAL)
public class MultiNewsModel {
  @ChildResource
  private List<NewsItem> newsList = new ArrayList<>();
  public List<NewsItem> getNewsList() {
    return newsList;
  }
  public static class NewsItem {
    @ValueMapValue
    private String newsTitle;
    @ValueMapValue
    private String newsSource;
    public String getNewsTitle() { return newsTitle; }
    public String getNewsSource() { return newsSource; }
  }
}
```

Step 3: Update newsComponent.html

```
<div class="cop-news-component">
  <h2>${news.newsTitle}</h2>
  <p>${news.newsDetail}</p>
  <p class="date">Published Date: ${news.publishedDate}</p>
  <p>Source: ${news.newsSource}</p>
</div>
```

Screenshot:



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

Step 1: Create Multi-Field Component

- Create a component named multiNewsComponent under/apps/myTraining/components/.
- Inside cq:dialog, define a multi-field.

Code:

```
<content jcr:primaryType="nt:unstructured"
    sling:resourceType="granite/ui/components/coral/foundation/container">
  <items jcr:primaryType="nt:unstructured">
    <newsItems jcr:primaryType="nt:unstructured"
        sling:resourceType="granite/ui/components/coral/foundation/form/multifield"
        fieldLabel="News Items">
      <field jcr:primaryType="nt:unstructured"
        sling:resourceType="granite/ui/components/coral/foundation/form/container">
       <items jcr:primaryType="nt:unstructured">
        <title jcr:primaryType="nt:unstructured"
          sling:resourceType="granite/ui/components/coral/foundation/form/textfield"
          fieldLabel="News Title"
          name="./title"/>
        <source jcr:primaryType="nt:unstructured"
          sling:resourceType="granite/ui/components/coral/foundation/form/textfield"
          fieldLabel="News Source"
          name="./source"/>
       </items>
      </field>
    </newsItems>
  </items>
</content>
```

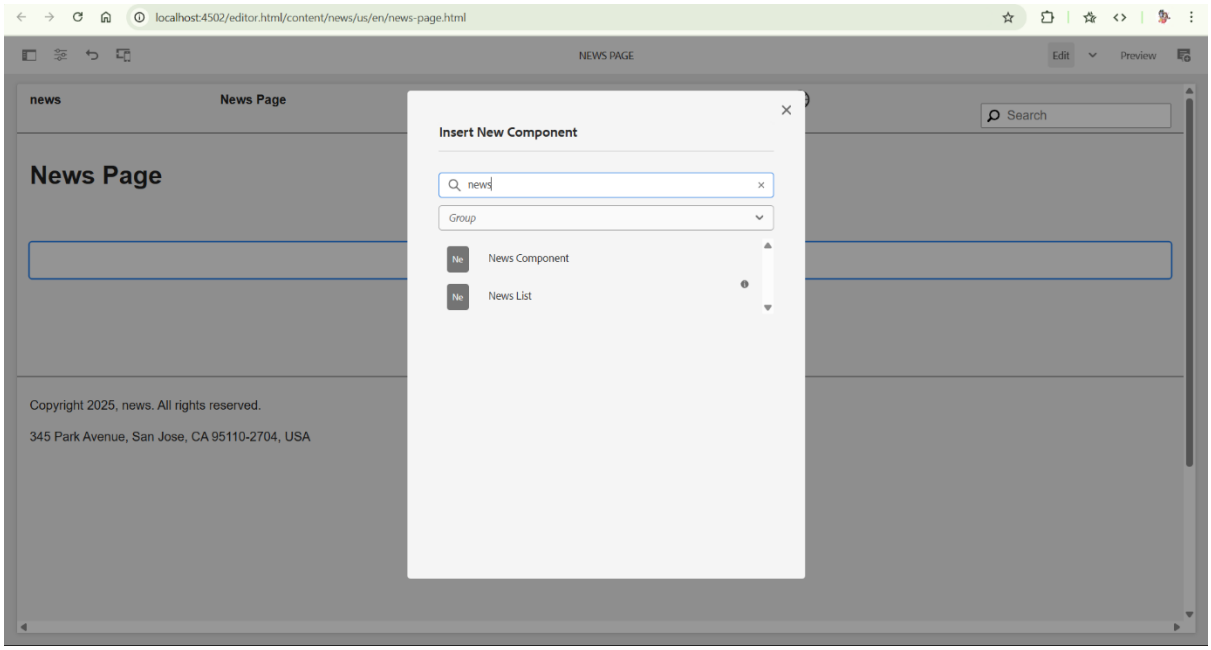
Step 2: Create Sling Model for Multi-Field Component

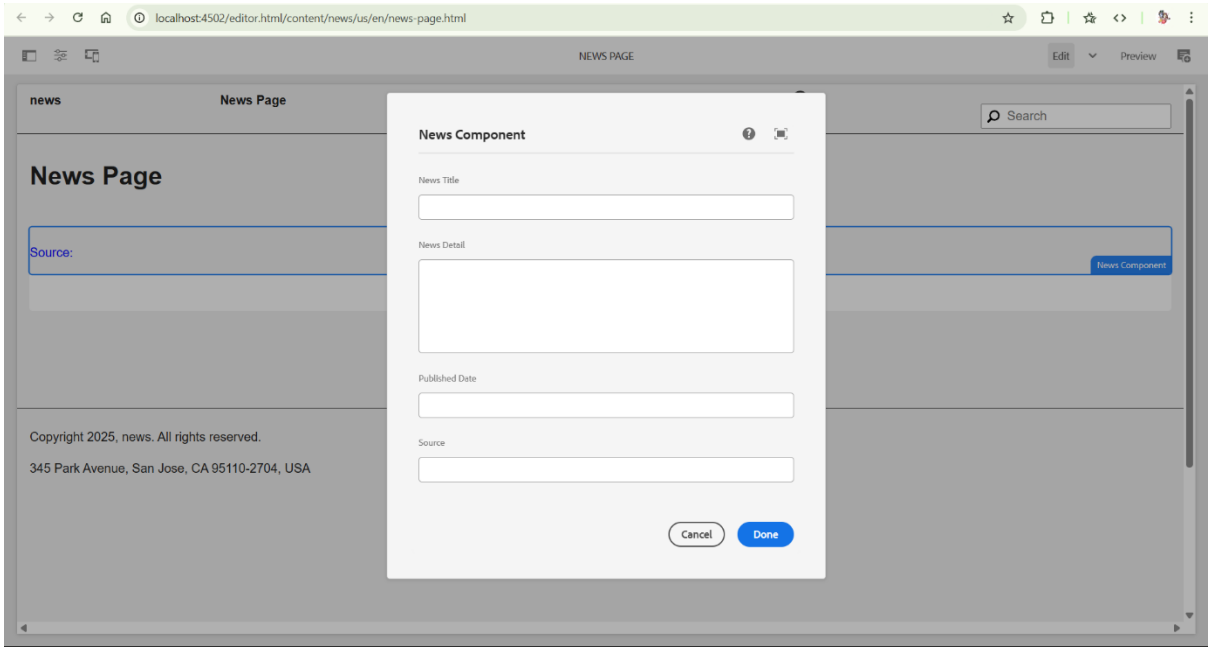
NewsItem.java

```
package com.news.core.models;
import org.apache.sling.api.resource.Resource;
import javax.inject.Inject;
import java.util.ArrayList;
import java.util.List;
```

```
import java.util.Optional;
import org.apache.sling.models.annotations.DefaultInjectionStrategy;
import org.apache.sling.models.annotations.Model;
@Model(adaptables = Resource.class, defaultInjectionStrategy = DefaultInjectionStrategy.OPTIONAL)
public class NewsListModel {
  @Inject
  private List<NewsItem> newsItems;

  public List<NewsItem> getNewsItems() {
    return Optional.ofNullable(newsItems).orElse(new ArrayList<>());
  }
  public static class NewsItem {
    @Inject
    private String title;

    @Inject
    private String source;

    public String getTitle() {
      return title;
    }
    public String getSource() {
      return source;
    }
  }
}
```

Screenshot:





3. Create Clientlibs for News Component

Steps:

• Navigate to /apps/myTraining/clientlibs/ and create a new folder newsClientLib.

• Add the following properties in newsClientLib:

- jcr:primaryType: cq:ClientLibraryFolder
- categories: [newsComponent]

• Inside newsClientLib, create css.txt and news.css.

• Add styles in news.css.

Code:

```
.cop-news-component .news-title {
  color: green;
}

.cop-news-component .news-detail {
  color: black;
}

.cop-news-component .news-date {
  color: black;
}

.cop-news-component .news-source {
  color: blue
}
```

Code:

```
<sly data-sly use.clientlib="/libs/granite/sightly/templates/clientlib.html"/>

<sly data-sly-call="${clientlib.css @ categories='newsComponent'}"/>
```

4. Add Custom Style Name to News Component

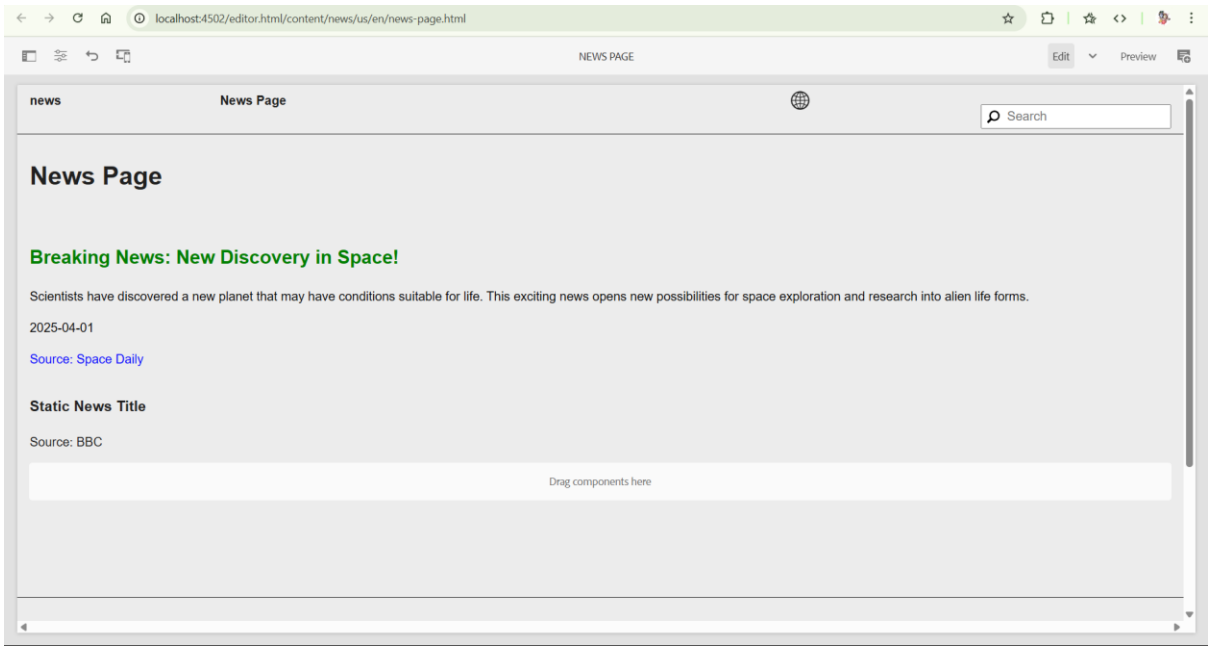
Steps:

1. Navigate to newsComponent → cq:editConfig.
2. Add the following property.

Code:

```
<cq:styleGroups

jcr:primaryType="nt:unstructured">

<newsStyle

    jcr:primaryType="nt:unstructured"

    cq:styleGroupLabel="News Styles">

    <cop-news-component

      jcr:primaryType="nt:unstructured"

      cq:styleLabel="News Component Style"/>

  </newsStyle>

</cq:styleGroups>
```

Screenshot:



5. Base Page Component with Metadata

Steps:

1. Modify basepage.html to include OpenGraph meta tags.
2. Configure the base component to pull metadata dynamically.
3. Deploy and verify metadata inclusion in the page source.

Code:

```
<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>
```

6. Custom Page Properties for Global Properties

Steps:

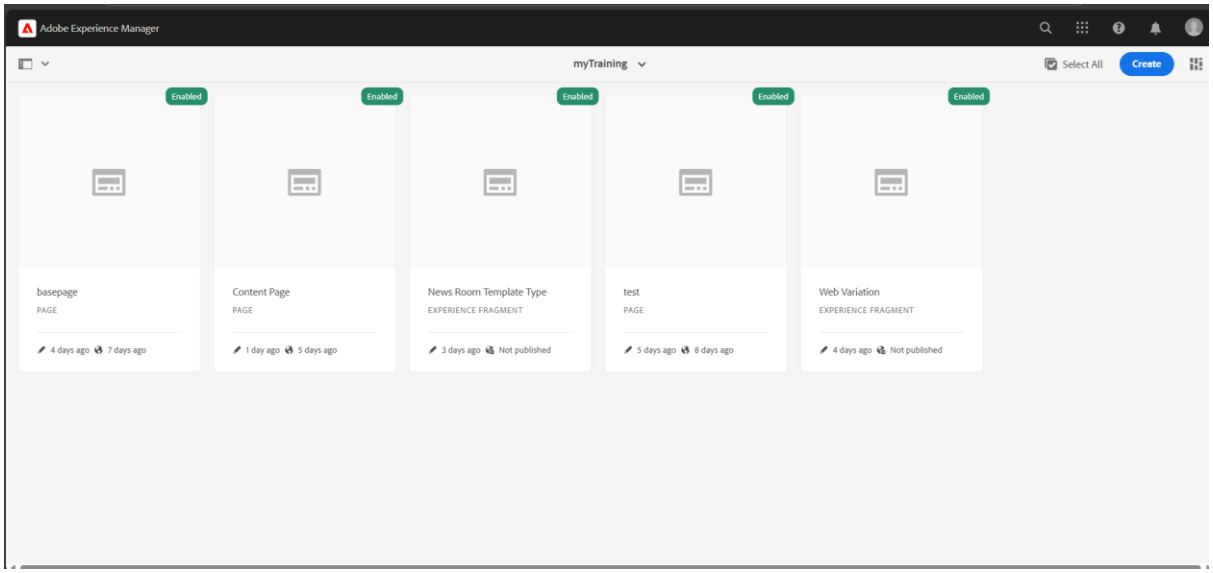
1. Modify _cq_dialog/.content.xml to include global properties fields.
2. Configure the page dialog to accept og:title, og:description, and og:image.
3. Deploy and verify the global properties in page properties.

Code:

```
<jcr:root
  xmlns:cq="http://www.day.com/jcr/cq/1.0"
  xmlns:jcr="http://www.jcp.org/jcr/1.0"
  xmlns:sling="http://sling.apache.org/jcr/sling/1.0"
  jcr:primaryType="cq:Dialog"
  title="Global Properties"
```

```
xtype="dialog">
<items jcr:primaryType="cq:Panel">
  <items jcr:primaryType="cq:WidgetCollection">
    <ogTitle
      jcr:primaryType="nt:unstructured"
      fieldLabel="OG Title"
      name="./og:title"
      xtype="textfield"/>
    <ogDescription
      jcr:primaryType="nt:unstructured"
      fieldLabel="OG Description"
      name="./og:description"
      xtype="textfield"/>
    <ogImage
      jcr:primaryType="nt:unstructured"
      fieldLabel="OG Image Path"
      name="./og:image"
      xtype="pathfield"/>
  </items>
 </items>
</jcr:root>
```

Screenshot:



7. What is extraClientLib and How to Add It to Multi-Field Component?

extraClientLib is used to include additional client-side libraries dynamically. To add it in multiNewsComponent, update cq:editConfig.

Steps:

1. Modify _cq_dialog/.content.xml to include extra clientlibs.
2. Update news.html to reference extra clientlibs.

3. Deploy and verify the applied styles.

Code:

- _cq_dialog/.content.xml (Extra clientlibs configuration)

```
<extraClientLibs
  jcr:primaryType="nt:unstructured"
  extraClientLibs="[myproject.clientlibs.custom]"/>
```

- news.html (Include extra clientlibs)

```
<sly data-sly-use.clientLib="/libs/granite/sightly/templates/clientlib.html">
<sly data-sly-call="${clientLib.all @ categories='myproject.clientlibs.custom'}"/></sly>
```