AEM Training - Day 5 Assignment (24-03-2025)

Creating News Articles

Description:

We will be creating structured News Articles within AEM, organized under a dedicated section.

Steps:

- Navigate to /content/us/en/news in AEM.

- Create five distinct news pages, each with:

    o Headline (<h2>)

    o Article Content (<p>)

    o Publication Date

Adding the News Component

Description:

The News Component will be used to dynamically display the news articles.

Steps:

- Utilize the existing News Component to render the articles.

- Ensure the component includes:

    o Headline (<h2>) with unique styling.

    o Article Content (<p>) to display the details.

    o Publication Date with correct formatting.

    o Maintain a responsive, visually appealing layout.

Code Snippet (NewsComponent.html):

<div class="news-item">

 <h2>${newsModel.headline}</h2>

 <p>${newsModel.content}</p>

 ${newsModel.publicationDate}

```
</div>
```

Setting Up the Header

Description:

A Header Experience Fragment will be created to ensure seamless navigation across pages.

Steps:

- Create a Header Fragment.

- Include navigation links to:

    o News Section (linking to the news pages)

    o About Me page

    o Contact Us page

Configuring the Footer

Description:

A Footer Experience Fragment will be set up with multiple sections for better structure.

Sections:

1. **Latest News Section:**

    o Use a List Component to display recent news articles.

2. **About Me Section:**

    o Add a Text Component with journalist details.

3. **Contact Section:**

    o Include office address, phone number, or email information.

4. **Social Media Section:**

    o Use a List Component to display social media links.

Developing a Custom Service

Description:

A custom OSGi Service will be created to return a simple response.

Steps:

- Create an OSGi service class in Java.

- Inject the service into the News Component's Sling Model.

- Log the service output in the AEM logs.

Code Snippet (HelloWorldService.java):

```
@Designate(ocd = HelloWorldConfig.class)

@Component(service = HelloWorldInt.class)

public class HelloWorldService {

   public String getMessage() {

      return "Hello World from Custom Service";

   }

}
```

Implementing API Configuration
Description:
We will develop an API configuration to store third-party API URLs and fetch data.
Steps:

- Define an OSGi configuration for API URLs.

- Fetch JSON data from a sample API.

- Log the response in AEM logs.

Example API:
https://jsonplaceholder.typicode.com/posts

Screenshots:

☑ Select All   Create   III ⌄

☐ • us	>	☐ • en	>	☐ News-Room news-room	>	☐ • News news	>	☐ • News-1 news-1

☐ • About Us about-us   ☐ • News-2 news-2

☐ • Contact Us contact-us   ☐ • News-3 news-3

☐ • News-4 news-4

☐ • News-5 news-5

🔍 ⊞ ? 🔔 👤

☑ Select All   Create   ⚙   III ⌄

☐ 📁 en	>	☐ 📁 Site	>	☐ 🌐 Header	>	☐ • Header XF

☐ 🌐 Footer   >

☐ 🌐 Header XF   >

☐ 🌐 Footer XF   >

Content Tree   NEWS-1

⌄ Container   • News   • About Us

   Title: News-1

  ⌄ Container   News-1

    News Component   🔧

News   ? ⛶

News Headline   ⓘ
┌───┐
│ News Sample   │
└───┘

NEWS Detail Article   ⓘ
┌───┐
│ Sample Details of news   │
│   │
│   │
│   │
└───┘

News Image   ⓘ
┌──────────────────────────────────┬──────┐
│   │ 📁 │
└──────────────────────────────────┴──────┘

News Publish Date   ⓘ
┌──────────────────────────────────┬──────┐
│ 12 March, 2025   │ 📅 │
└──────────────────────────────────┴──────┘

News Titles