

Creating News Articles in AEM :

Steps to Create News Pages:

- Navigate to /content/us/en/news in AEM.
- Create five unique news pages, each containing:
 - A headline (<h2>) for the article title.
 - Article content (<p>) detailing the news story.
 - A publication date to indicate when the article was published.

Integrating the News Component:

The News Component will be utilized to dynamically present news articles in an organized manner.

Steps to Implement:

- Use the existing News Component to render the articles efficiently.
- Ensure the component includes:
 - A headline (<h2>) with distinct styling for emphasis.
 - Article content (<p>) to display detailed information.
 - A publication date, properly formatted for clarity.
- Maintain a responsive and visually engaging layout to enhance the user experience across different devices.

Code snippet:

```
<div class="news-item">  
  
  <h2>${newsModel.headline}</h2>  
  
  <p>${newsModel.content}</p>  
  
  <span>${newsModel.publicationDate}</span>  
  
</div>
```

This snippet defines the News Component, which dynamically displays news articles. The headline (<h2>) is styled distinctly, the content (<p>) presents the article details, and the publication date () is formatted appropriately.

Setting Up the Header Description:

To enable smooth navigation across the website, a Header Experience Fragment will be developed.

Implementation Steps:

- Create a Header Fragment to serve as a reusable navigation component.
- Add navigation links directing users to:
 - The News Section, linking to the news pages.
 - The About Me page for author details.
 - The Contact Us page for easy communication.

Setting Up the Footer Experience Fragment

A **Footer Experience Fragment** will be designed with well-structured sections for enhanced organization and usability.

Footer Sections:

- **Latest News:**
 - Implement a **List Component** to showcase recent news articles.
- **About Me:**
 - Add a **Text Component** containing details about the journalist.
- **Contact Information:**
 - Include essential details such as **office address, phone number, and email** for easy communication.
- **Social Media Links:**
 - Utilize a **List Component** to display links to relevant social media profiles.

Building a Custom OSGi Service

A **custom OSGi service** will be developed to generate a simple response.

Implementation Steps:

- Develop an **OSGi service class** using Java.
- Integrate the service into the **News Component's Sling Model** through dependency injection.
- Capture and log the **service output** in the AEM logs for validation.

Code snippet :

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

@Component(service = HelloWorldInt.class)

public class HelloWorldService {

public String getMessage() {

return "Hello World from Custom Service"; }

}
```

Setting Up API Configuration

An API configuration will be implemented to store third-party API URLs and retrieve data dynamically.

Steps to Implement:

- Create an **OSGi configuration** to manage API endpoint URLs.
- Retrieve **JSON data** from a sample external API.
- Log the **API response** in AEM logs for monitoring and debugging.

Screenshots :

