

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

### Creating News Articles:

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:
- Headline (<h2>)
- Article Content (<p>)
- Publication Date

### Adding the News Component:

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:
  - Headline (<h2>) with unique styling.
  - Article Content (<p>) to display the details.
  - Publication Date with correct formatting.
  - Maintain a responsive, visually appealing layout.

### Code Snippet (NewsComponent.html):

```
<div class="news-item">  
  
  <h2>${newsModel.headline}</h2>  
  
  <p>${newsModel.content}</p>  
  
  <span>${newsModel.publicationDate}</span>  
  
</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:
  - News Section (linking to the news pages)
  - About Me page
  - Contact Us page

**Configuring the Footer:**

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

**Latest News Section:**

Use a List Component to display recent news articles.

**About Me Section:**

Add a Text Component with journalist details.

**Contact Section:**

Include office address, phone number, or email.

**Social Media Section:**

Use a List Component to display social media links.

**Developing a Custom Service:**

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:

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:





