

**Date: 24/03/2025**

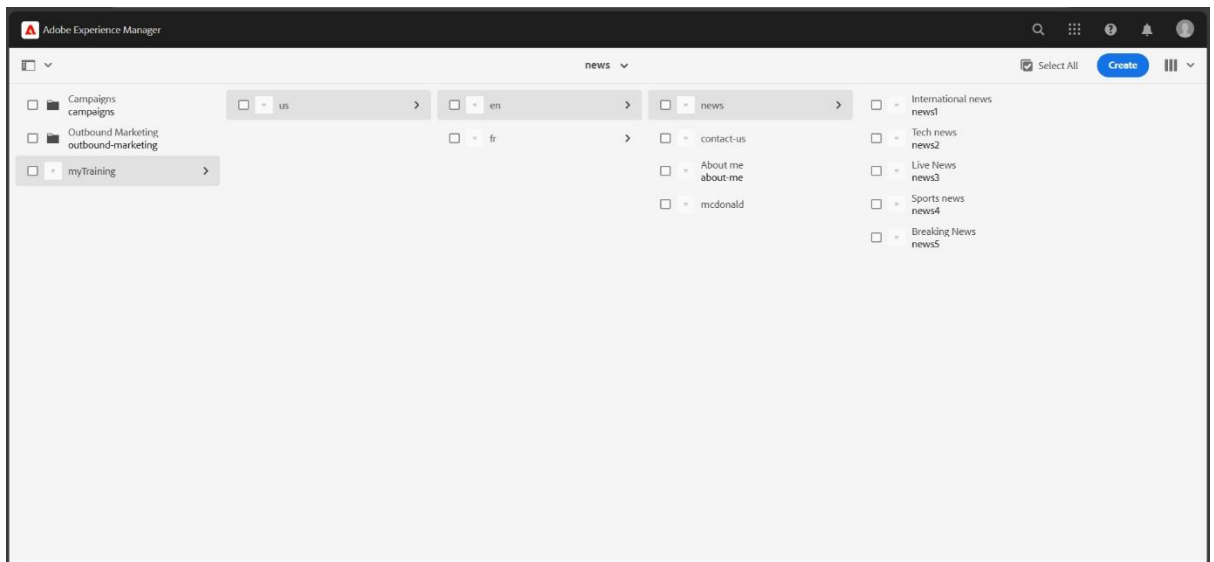
## **Table of Contents**

1. Create 5 News Article Pages
2. Use News Component
3. Create Header Experience Fragment
4. Create Footer Experience Fragment
5. Create Custom Service
6. Create Custom Configuration

### **1. Create 5 News Article Pages**

#### **Steps:**

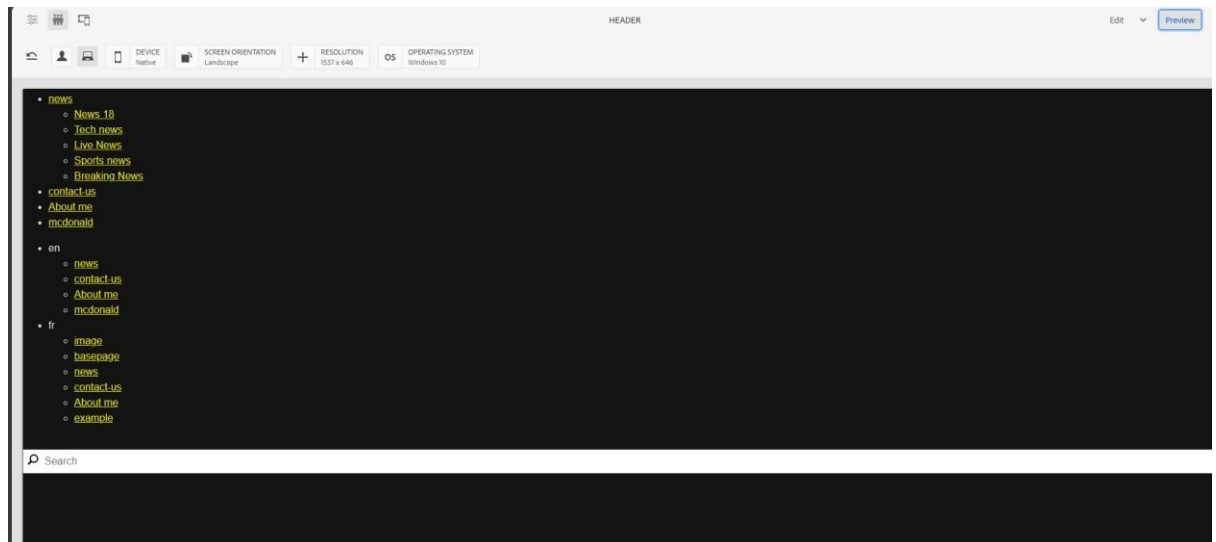
1. Access the CMS Dashboard: Log into the content management system (CMS) and navigate to the "Pages" section.
2. Create a New Page: Click on "Create New Page" and select the "News Article" template.
3. Enter Article Details:
  - Add a title that reflects the article's content.
  - Write a body with relevant information and formatting.
  - Upload and insert images to enhance readability.
  - Add tags and metadata for better searchability.
4. Apply Formatting & Styling:
  - Use headers, bullet points, and paragraphs for structured content.
  - Embed videos or links if necessary.
5. Preview & Optimize:
  - Check for grammar and spelling mistakes.
  - Ensure the page is mobile-responsive.
6. Publish the Article: Save and publish the article.
7. Repeat the Process: Follow the same steps to create five unique articles.



## 2. Use News Component

### Steps:

1. Navigate to the CMS Editor: Open the editor for a newly created or existing News Article Page.
2. Locate the News Component: Find the "News Component" in the component panel.
3. Drag & Drop: Place the News Component in the desired section of the page.
4. Configure the Component:
  - Set the number of articles to display.
  - Choose layout options (grid, list, or carousel view).
  - Define sorting (latest, most popular, or custom order).
5. Enable Dynamic Content Fetching (if required): Connect to an API or database to fetch news dynamically.
6. Save & Publish Changes: Ensure the component appears correctly on the published page.



### 3. Create Header Experience Fragment

#### Steps:

1. Navigate to Experience Fragments: Go to the "Experience Fragments" section in the CMS.
2. Create a New Fragment: Click "Create New Fragment" and select the "Header Template."
3. Add Header Elements:
  - Logo: Insert and align the company logo.
  - Navigation Menu: Add links to main sections (Home, News, Contact, etc.).
  - Search Bar: Enable a search feature if applicable.
  - User Profile/Login: Include login or signup options if required.
4. Customize Styling: Adjust fonts, colors, and spacing to match the brand identity.
5. Save & Publish the Header Fragment.
6. Apply to Relevant Pages: Reference the Header Fragment in templates or master pages.

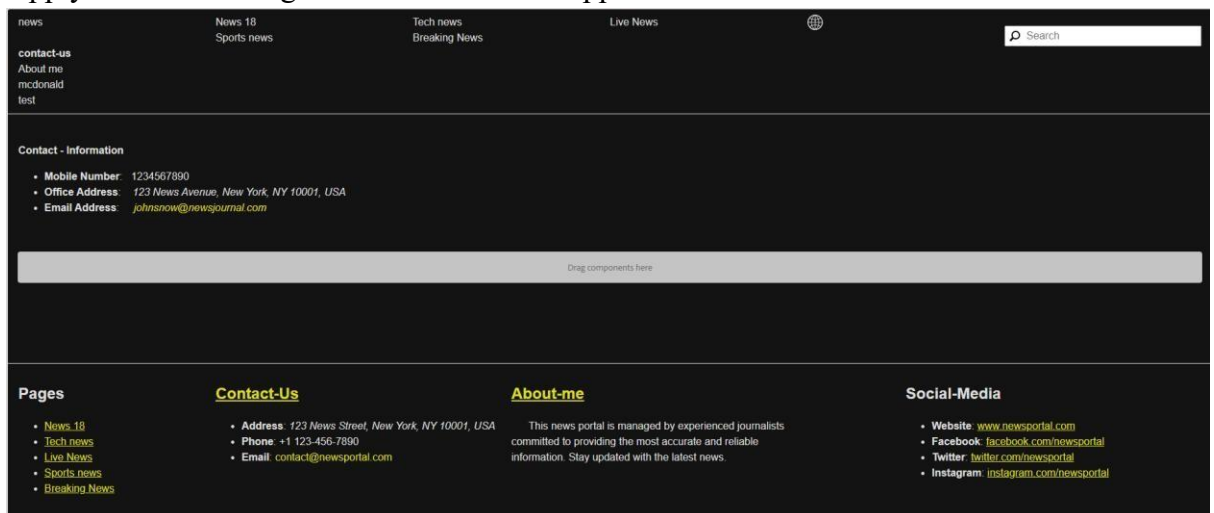


### 4. Create Footer Experience Fragment

#### Steps:

1. Go to Experience Fragments: Navigate to "Experience Fragments" in the CMS.

2. Create a New Fragment: Click "Create New Fragment" and choose the "Footer Template."
3. Add Footer Elements:
  - Company Information: Name, address, and contact details.
  - Quick Links: Shortcuts to important pages (Privacy Policy, About Us, etc.).
  - Social Media Links: Icons with external links to social media platforms.
  - Copyright Notice: Add a statement to indicate ownership.
4. Style the Footer: Apply consistent styling with the header.
5. Save & Publish the Footer Fragment.
6. Apply to Relevant Pages: Ensure the footer appears across the website.



## 5. Create Custom Service

### Steps:

1. Open Development Environment: Use an Integrated Development Environment (IDE) such as Visual Studio Code or IntelliJ.
2. Create a New Service Class:
  - Define a new service in the backend framework (Java, Node.js, Python, etc.).
  - Name it appropriately (e.g., NewsService).
3. Implement Business Logic:
  - Fetch data from a database or API.
  - Process and return results based on requirements.
  - Handle errors and logging.
4. Test the Service:

```
02.04.2025 09:23:06.845 *INFO* [[0:0:0:0:0:0:1] [1743567786653] GET /content/myTraining/us/en/img/jcr:content/root/container/container/news.html HTTP/1.1] com.myTraining.core.models.NewsComponentModel News Component says: Hello World from AEM Service!
```

- Write and execute unit tests.
- Debug and fix any issues found.

#### 5. Deploy to the Server:

- Integrate with the application.
- Push to the production server after testing.

### 6. Create Custom Configuration

#### Steps:

1. Access Configuration Files: Navigate to the configuration directory in the CMS or backend framework.
2. Create a New Configuration Entry:
  - Define the required settings.
  - Use environment-specific variables where necessary.
3. Apply Configurations to Components: Ensure new settings are properly referenced in the application.
4. Validate & Test Configurations:
  - Check for syntax errors.
  - Perform tests to ensure functionality.
5. Save & Deploy Changes: Commit changes and update the live environment.