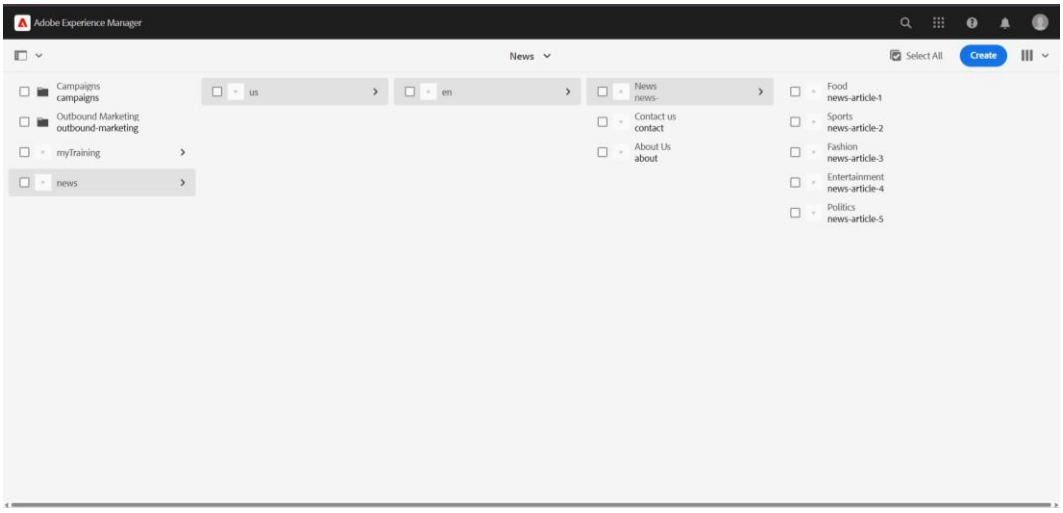
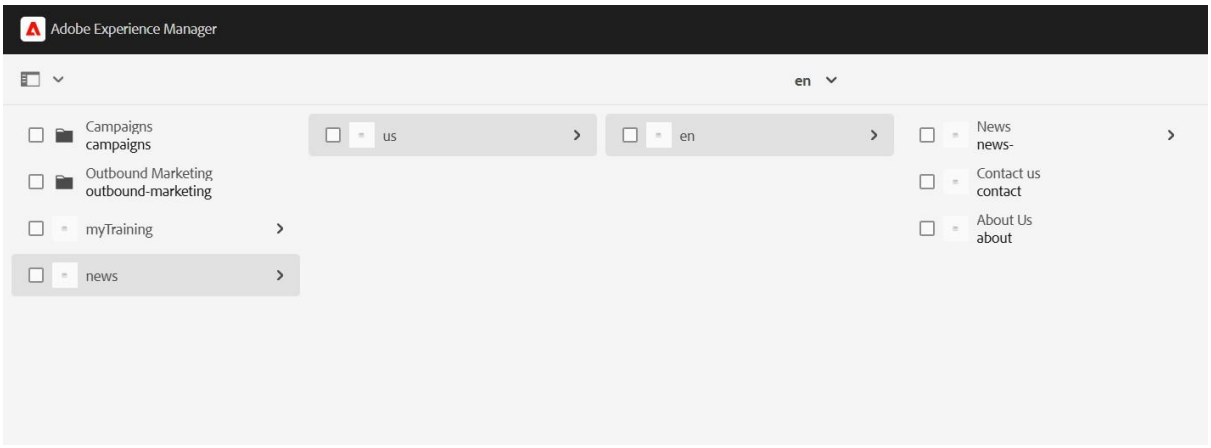
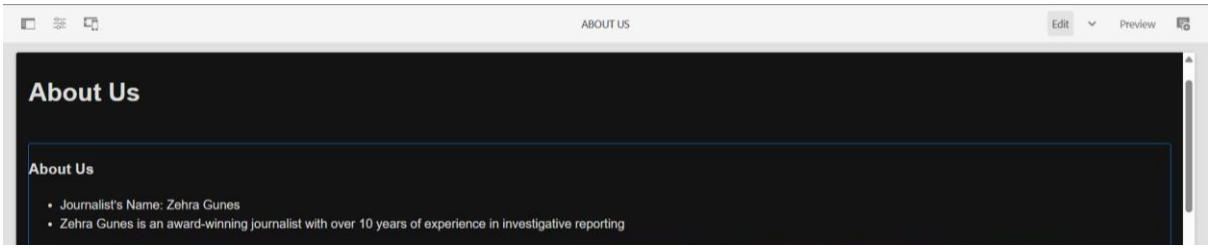
1. **Create 5 News Article Pages** under /content/us/en/news

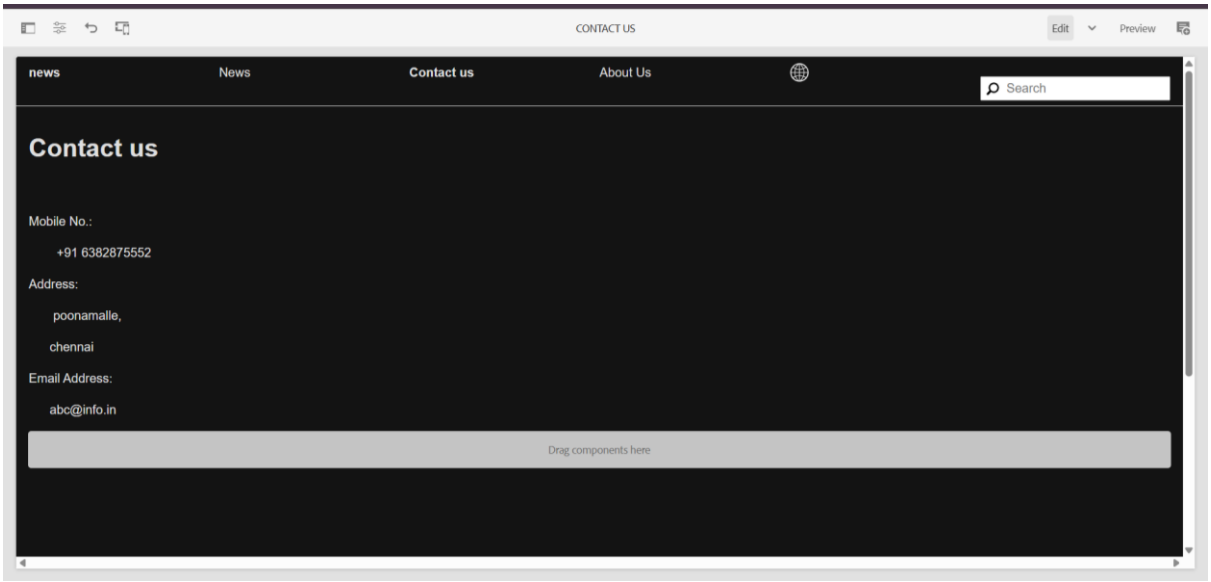


2. Create Header Experience fragment for header This will include: 1. News Menu – Links to the 5 news article pages you created. 2. Contact Us Page – With office address, email, or mobile number

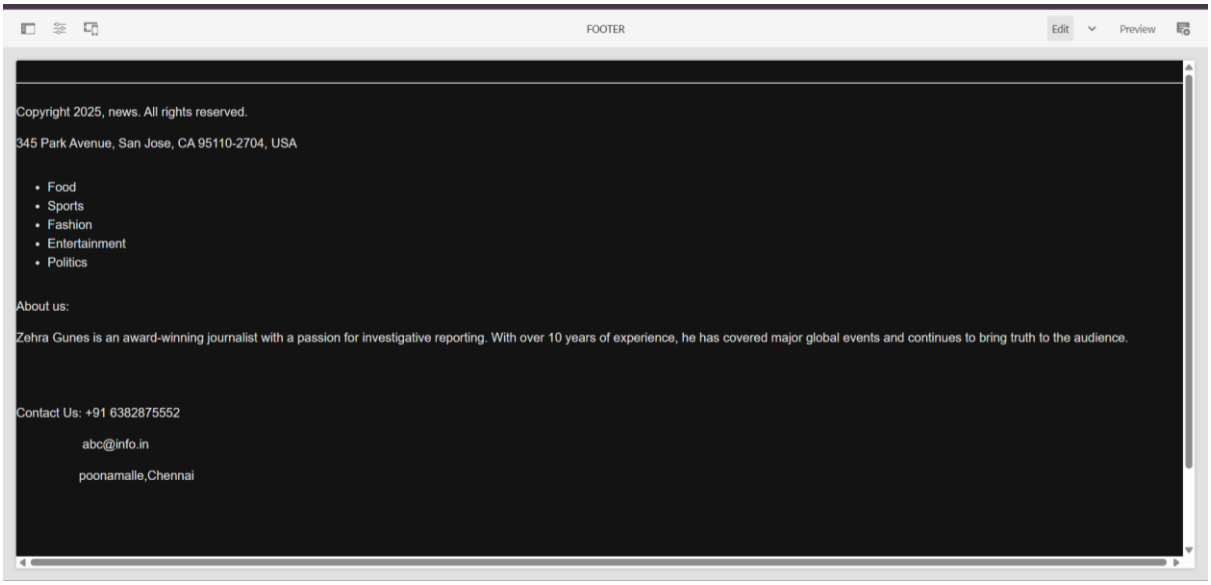




4. Contact Us



5. Footer



Create Custom OSGi Service to Print “Hello World”

Step 1: Service Interface (Java)

```
package com.aem.core.services;

public interface HelloWorldService {
    String getMessage();
}
```

Step 2: Service Implementation

```
package com.aem.core.services.impl;

import com.aem.core.services.HelloWorldService;
import org.osgi.service.component.annotations.Component;

@Component(service = HelloWorldService.class)
public class HelloWorldServiceImpl implements HelloWorldService {
    @Override
    public String getMessage() {
```

```
        return "Hello World";
    }
}
```

Step 3: Inject and Log in Sling Model

```
package com.aem.core.models;

import com.aem.core.services.HelloWorldService;
import org.apache.sling.api.SlingHttpServletRequest;
import org.apache.sling.models.annotations.DefaultInjectionStrategy;
import org.apache.sling.models.annotations.Model;
import org.osgi.service.component.annotations.Reference;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

import javax.inject.Inject;

@Model(adaptables = SlingHttpServletRequest.class, defaultInjectionStrategy =
DefaultInjectionStrategy.OPTIONAL)
public class NewsModel {

    private static final Logger LOG = LoggerFactory.getLogger(NewsModel.class);

    @Inject
    private HelloWorldService helloWorldService;

    @PostConstruct
    protected void init() {
        if (helloWorldService != null) {
            LOG.info("Message from HelloWorldService: {}", helloWorldService.getMessage());
        }
    }
}
```

Custom OSGi Configuration to Fetch JSON from 3rd Party API

Step 1: Create Config Interface

```
package com.aem.core.config;

import org.osgi.service.metatype.annotations.AttributeDefinition;
import org.osgi.service.metatype.annotations.ObjectClassDefinition;

@ObjectClassDefinition(name = "Third Party API Configuration")
public @interface ThirdPartyApiConfig {

    @AttributeDefinition(name = "API URL")
    String apiUrl() default "https://jsonplaceholder.typicode.com/posts";
}
```

Step 2: Create Service to Call API and Log JSON

```
java
CopyEdit
package com.aem.core.services.impl;

import com.aem.core.config.ThirdPartyApiConfig;
import org.apache.commons.io.IOUtils;
import org.osgi.service.component.annotations.Activate;
import org.osgi.service.component.annotations.Component;
```

```
import org.osgi.service.component.annotations.Modified;
import org.osgi.service.metatype.annotations.Designate;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

import javax.annotation.PostConstruct;
import java.io.InputStream;
import java.net.URL;
import java.nio.charset.StandardCharsets;

@Component(immediate = true)
@Designate(ocd = ThirdPartyApiConfig.class)
public class ThirdPartyApiService {

    private static final Logger LOG = LoggerFactory.getLogger(ThirdPartyApiService.class);
    private String apiUrl;

    @Activate
    @Modified
    protected void activate(ThirdPartyApiConfig config) {
        this.apiUrl = config.apiUrl();
    }

    @PostConstruct
    public void fetchDataFromApi() {
        try (InputStream inputStream = new URL(apiUrl).openStream()) {
            String json = IOUtils.toString(inputStream, StandardCharsets.UTF_8);
            LOG.info("Fetched JSON Data: {}", json);
        } catch (Exception e) {
            LOG.error("Error fetching data from API", e);
        }
    }
}
```