26/03/2025

Tasks:

1. Create SampleServlet using SlingAllMethodServlet

2. Create CreatePageServlet using SlingSafeMethodServlet

3. Create AEM pages dynamically using servlet

4. Use Page Manager APIs for page creation

5. Create SearchServlet using PredicateMap for content search

1. Create SampleServlet using SlingAllMethodServlet

Steps:

1. **Create Java Class:**

    o   Extend SlingAllMethodsServlet to support both GET and POST requests.

2. **Register Servlet using Resource Type:**

    o   Use @SlingServletResourceTypes annotation.

    o   Define resourceTypes and methods properties.

3. **Override doGet and doPost Methods:**

    o   Implement logic for handling GET and POST requests.

    o   Print sample responses to verify execution.

4. **Deploy and Test Servlet:**

    o   Register servlet in AEM.

    o   Access the servlet via the defined resource type.

Code:

package com.myTraining.core.servlets;

import com.day.cq.commons.jcr.JcrConstants;

import org.apache.sling.api.SlingHttpServletRequest;

import org.apache.sling.api.SlingHttpServletResponse;

import org.apache.sling.api.resource.Resource;

import org.apache.sling.api.servlets.HttpConstants;

```java
import org.apache.sling.api.servlets.SlingAllMethodsServlet;

import org.apache.sling.servlets.annotations.SlingServletResourceTypes;

import org.osgi.service.component.annotations.Component;

import org.osgi.service.component.propertytypes.ServiceDescription;


import javax.servlet.Servlet;

import javax.servlet.ServletException;

import java.io.IOException;


@Component(service = { Servlet.class })
@SlingServletResourceTypes(
    resourceTypes="myTraining/components/page",
    methods=HttpConstants.METHOD_GET,
    extensions="txt")
@ServiceDescription("Simple Demo Servlet")
public class SimpleServlet extends SlingAllMethodsServlet {
    private static final long serialVersionUID = 1L;


    @Override
    protected void doGet(final SlingHttpServletRequest req,
                final SlingHttpServletResponse resp) throws ServletException, IOException {
        final Resource resource = req.getResource();

        resp.setContentType("text/plain");

        resp.getWriter().write("Title = " +
resource.getValueMap().get(JcrConstants.JCR_TITLE));
    }
}
```
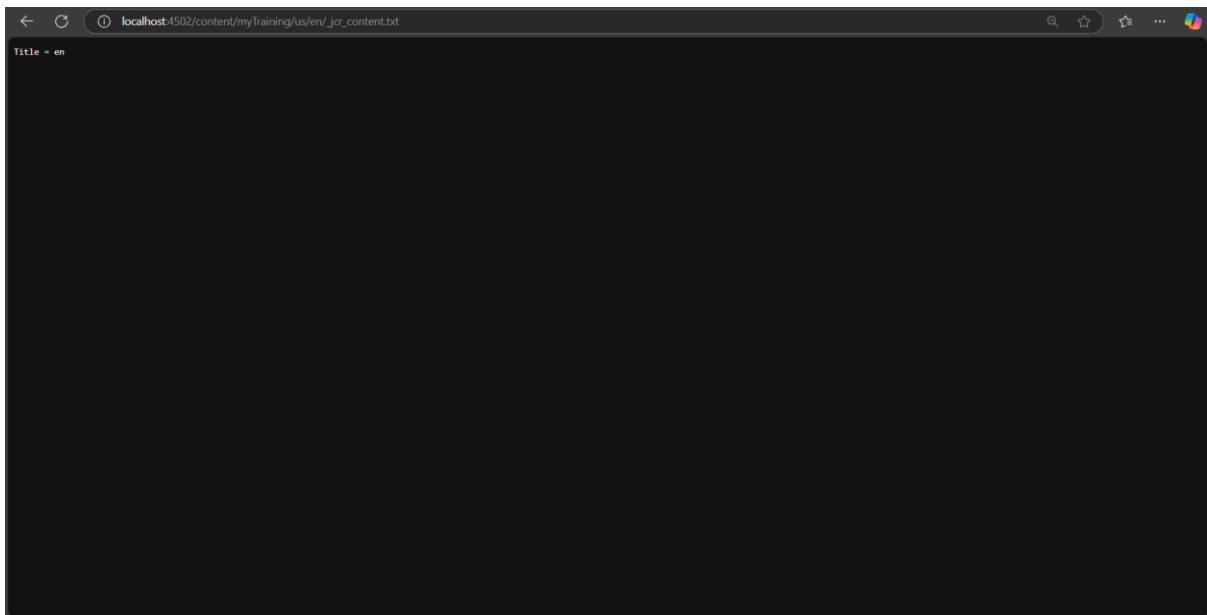
2. Create CreatePageServlet using SlingSafeMethodServlet

Steps:

1. **Create Java Class:**

   o   Extend SlingSafeMethodsServlet to handle safe HTTP methods.

2. **Register Servlet using Path:**

   o   Use @SlingServletPaths annotation.

   o   Define paths and methods properties.

3. **Implement Logic to Create Page:**

   o   Retrieve pageName parameter from request.

   o   Use PageManager to create a new page under /content.

   o   Use an existing template (/conf/template).

4. **Deploy and Verify Servlet:**

   o   Access servlet via /bin/createpage?pageName=samplePage.

   o   Check if the page is created in AEM under /content/samplePage.

Code:

package com.myTraining.core.servlets;

import com.day.cq.wcm.api.Page;

import com.day.cq.wcm.api.PageManager;

```java
import com.day.cq.wcm.api.WCMException;

import org.apache.sling.api.SlingHttpServletRequest;

import org.apache.sling.api.SlingHttpServletResponse;

import org.apache.sling.api.servlets.SlingSafeMethodsServlet;

import org.osgi.service.component.annotations.Component;

import org.osgi.service.component.annotations.Reference;


import javax.servlet.Servlet;

import javax.servlet.ServletException;

import java.io.IOException;


@Component(

    service = Servlet.class,

    property = {

        "sling.servlet.paths=/bin/createPage",

        "sling.servlet.methods=GET"

    }

)
public class CreatePageServlet extends SlingSafeMethodsServlet {

    private static final String PARENT_PATH = "/content/myTraining";

    private static final String TEMPLATE_PATH =
"/conf/myTraining/settings/wcm/templates/page-content";


    @Reference

    private org.apache.sling.api.resource.ResourceResolverFactory resourceResolverFactory;


    @Override

    protected void doGet(SlingHttpServletRequest request, SlingHttpServletResponse
response) throws ServletException, IOException {

        String pageName = request.getParameter("pageName");
```

```
    if (pageName == null || pageName.isEmpty()) {

        response.getWriter().write("Page name is required!");

        return;

    }


    org.apache.sling.api.resource.ResourceResolver resolver =
request.getResourceResolver();

    PageManager pageManager = resolver.adaptTo(PageManager.class);


    if (pageManager != null) {

        try {

            Page newPage = pageManager.create(PARENT_PATH, pageName,
TEMPLATE_PATH, pageName);

            response.getWriter().write("Page created successfully at: " + newPage.getPath());

        } catch (WCMException e) {

            response.getWriter().write("Error creating page: " + e.getMessage());

        }

    } else {

        response.getWriter().write("PageManager is null. Unable to create page.");

    }

  }

}
```

3. Create AEM pages dynamically using servlet

Steps:

1. **Accept User Input:**
   - Pass pageName as a request parameter.

2. **Invoke CreatePageServlet:**
   - Call /bin/createpage?pageName=dynamicPage.

3. **Verify Page Creation:**

   o Navigate to /content/dynamicPage in AEM's Site Admin.

4. Use Page Manager APIs for page creation

Best Practices:

- **Ensure Service User Permissions:**

  o Use system users with proper permissions.

- **Use Templates & Components:**

  o Assign pages a meaningful template.

- **Handle Naming Conflicts:**

  o Verify if the page already exists before creating it.

5. Create SearchServlet using PredicateMap for content search

Steps:

1. **Create Java Class:**

   o Extend SlingSafeMethodsServlet to ensure secure operations.

2. **Register Servlet using Path:**

   o Use @SlingServletPaths annotation to define endpoint /bin/searchcontent.

3. **Use Query Builder API:**

   o Construct a PredicateMap to search content.

   o Set path, type, and fulltext search parameters.

4. **Process Search Results:**

   o Retrieve results from QueryBuilder.

   o Return formatted search results.

Found page: /content/myTraining/us/in_id/news2
Found page: /content/myTraining/us/en/news2
Found page: /content/myTraining/us/in_id/newlive/news2