

ASSIGNMENT (25.03.2025)

1. Create SampleServlet extend the SlingAllMethod Servlet and register it using resourceType
2. Create CreatePageServlet extend the SlingSafeMethod Servlet and register it using path
- 3 Take page name from user and create pages in aem using above servlet
4. Use page Manager Apis for above task.
5. Create one SearchServlet to search the content using PredicateMap to search the content from pages

```
import javax.servlet.ServletException;
import javax.servlet.ServletException;
import java.io.IOException;

@Component(service = Servlet.class)
@SlingServletResourceTypes(
    resourceTypes = "myTraining/components/page",
    methods = { "GET", "POST" },
    extensions = "json"
)

public class SampleServlet extends SlingAllMethodsServlet {

    @Override 1 usage
    protected void doGet(SlingHttpServletRequest request, SlingHttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("application/json");
        response.getWriter().write("{\"message\": \"GET request processed\"}");
    }

    @Override no usages
    protected void doPost(SlingHttpServletRequest request, SlingHttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("application/json");
        response.getWriter().write("{\"message\": \"POST request processed\"}");
    }
}
```

org.apache.sling.api

@ProviderType

public interface SlingHttpServletRequest
extends javax.servlet.http.HttpServletRequest, org.apache.sling.api.adapter.Adaptable

Maven: com.adobe.aem:uber-jar:apis:6.5.5 (uber-jar-6.5.5-apis.jar)

Activate Windows

```

@Component(service = Servlet.class,
    property = {
        ServletResolverConstants.SLING_SERVLET_PATHS + "/bin/createPage"
    })
public class CreatePageServlet extends SlingSafeMethodsServlet {

    @Override 1 usage
    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;
        }

        ResourceResolver resolver = request.getResourceResolver();
        PageManager pageManager = resolver.adaptTo(PageManager.class);

        if (pageManager != null) {
            try {
                Page newPage = pageManager.create("/content", pageName, "/apps/example/templates/basic", 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 could not be initialized.");
        }
    }
}

```

Activate W
Go to Settings

```

        StringBuilder resultOutput = new StringBuilder("Search Results:\n");

        try {
            List<Hit> hits = result.getHits();
            for (Hit hit : hits) {
                resultOutput.append(hit.getPath()).append("\n");
            }
        } catch (Exception e) {
            response.getWriter().write("Error fetching search results: " + e.getMessage());
            return;
        }

        response.getWriter().write(resultOutput.toString());
    } else {
        response.getWriter().write("QueryBuilder could not be initialized.");
    }
}
}

```

```

import org.osgi.service.component.annotations.Component;
import org.apache.sling.api.servlets.ServletResolverConstants;
import org.apache.sling.api.resource.ResourceResolver;

import javax.servlet.Servlet;
import javax.servlet.ServletException;
import java.io.IOException;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

@Component(service = Servlet.class,
    property = {
        ServletResolverConstants.SLING_SERVLET_PATHS + "/bin/searchContent"
    })
public class SearchServlet extends SlingSafeMethodsServlet {

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

        String searchText = request.getParameter("searchText");

        if (searchText == null || searchText.isEmpty()) {
            response.getWriter().write("Search text is required.");
            return;
        }

        ResourceResolver resolver = request.getResourceResolver();
        QueryBuilder queryBuilder = resolver.adaptTo(QueryBuilder.class);

        if (queryBuilder != null) {

```