

1. Create SampleServlet extend the SlingAllMethod Servlet and register it using resourceType

```
4      import org.osgi.service.component.annotations.Component;
5      import org.apache.sling.servlets.annotations.SlingServletResourceTypes; import org.slf4j.Logger;
6      import org.slf4j.LoggerFactory;
7
8      import javax.servlet.Servlet;
9      import javax.servlet.ServletException; import java.io.IOException;
10
11      @Component(service = Servlet.class) @SlingServletResourceTypes( new *
12          resourceTypes = "assignment/components/sample", methods = {"GET", "POST"},
13          extensions = "json"
14      )
15      public class SampleServlet extends SlingAllMethodsServlet {
16          private static final Logger LOG = LoggerFactory.getLogger(SampleServlet.class); 2 usages
17
18          @Override 1 usage new *
19      @+@ protected void doGet(SlingHttpServletRequest request, SlingHttpServletResponse response) throws ServletException,
20          IOException {
21              LOG.info("Sample Servlet GET Request"); response.setContentType("application/json"); response.getWriter().write("{}");
22          }
23
24          @Override no usages new *
25      @+@ protected void doPost(SlingHttpServletRequest request, SlingHttpServletResponse response) throws ServletException,
26          IOException {
27              LOG.info("Sample Servlet POST Request"); response.setContentType("application/json"); response.getWriter().write("{}");
28          }
29      }
```

2. Create CreatePageServlet extend the SlingSafeMethod Servlet and register it using path

```
7      import org.slf4j.LoggerFactory;
8
9      import javax.jcr.Session; import javax.servlet.Servlet;
10     import javax.servlet.ServletException; import java.io.IOException;
11
12     @Component(service = Servlet.class) @SlingServletPaths("/bin/createPage") new *
13     public class CreatePageServlet extends SlingSafeMethodsServlet {
14         private static final Logger LOG = LoggerFactory.getLogger(CreatePageServlet.class); 3 usages
15
16         @Override 1 usage new *
17     @+@ protected void doGet(SlingHttpServletRequest request, SlingHttpServletResponse response) throws ServletException,
18         IOException {
19         String pageName = request.getParameter("pageName"); if (pageName == null || pageName.isEmpty()) {
20             response.getWriter().write("Page name is required."); return;
21         }
22
23         try {
24             Session session = request.getResourceResolver().adaptTo(Session.class);
25             PageManager pageManager = request.getResourceResolver().adaptTo(PageManager.class);
26
27             if (pageManager != null) {
28                 Page newPage = pageManager.create("/content/us/en", pageName, "/conf/wknd/settings/wcm/templates/page.html");
29                 LOG.info("Page created successfully at: {}", newPage.getPath()); response.getWriter().write("Page created successfully.");
30             } else {
31                 LOG.error("PageManager is null"); response.getWriter().write("Error: PageManager is null.");
32             }
33         } catch (WCMException e) { LOG.error("Error creating page: ", e);
34             response.getWriter().write("Error creating page.");
35         }
36     }
37 }
```

3 Take page name from user and create pages in aem using above servlet

1. Open a browser.
2. Visit: <http://localhost:4502/bin/createPage?pageName=newsPage1>
3. It will create a page under /content/us/en/newsPage1.

4. Use page Manager Apis for above task.

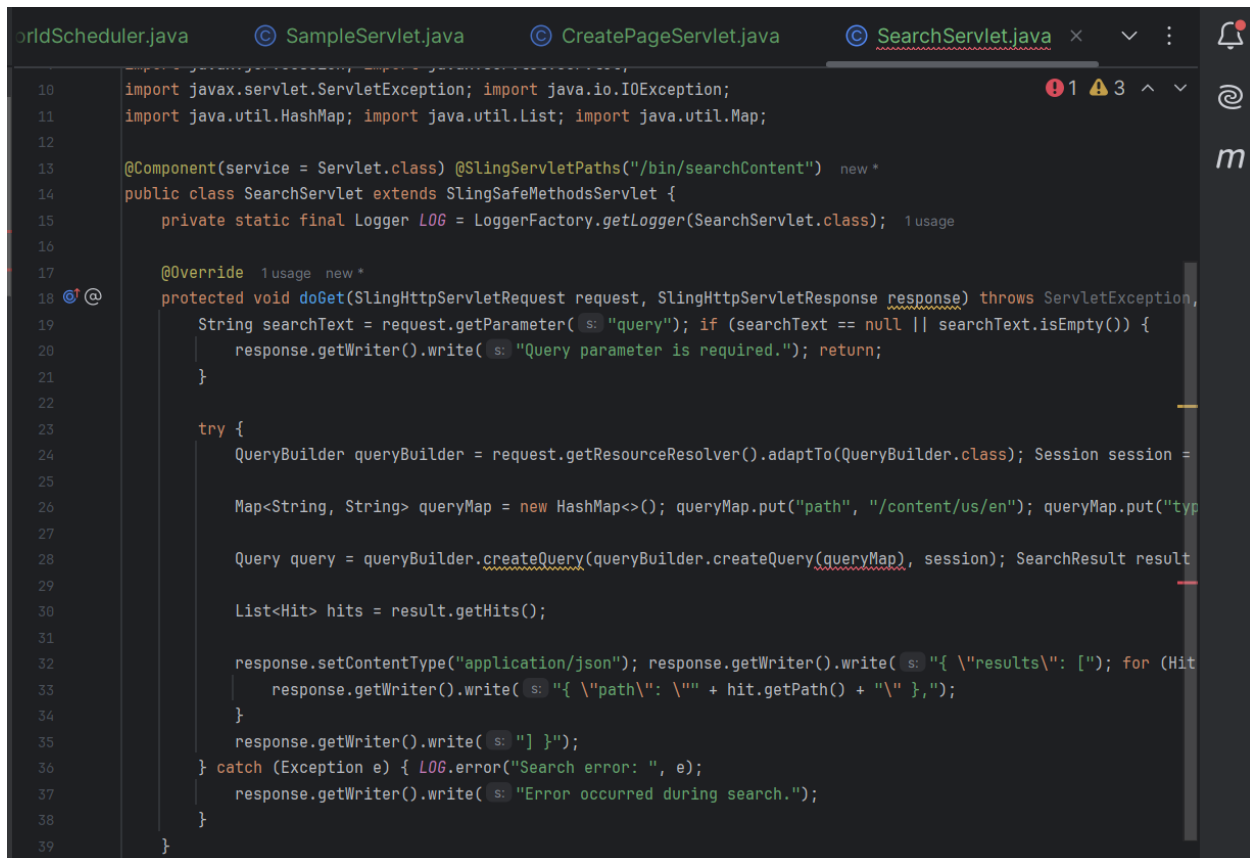
PageManager API Used in CreatePageServlet:

```
PageManager pageManager = request.getResourceResolver().adaptTo(PageManager.class);
```

```
Page newPage = pageManager.create("/content/us/en", pageName,  
"/conf/wknd/settings/wcm/templates/content-page", pageName);
```

It creates a new page under /content/us/en using an existing AEM template.

5. Create one SearchServlet to search the content using PredicateMap to search the content from pages



```
10 import javax.servlet.ServletException; import java.io.IOException;
11 import java.util.HashMap; import java.util.List; import java.util.Map;
12
13 @Component(service = Servlet.class) @SlingServletPaths("/bin/searchContent") new *
14 public class SearchServlet extends SlingSafeMethodsServlet {
15     private static final Logger LOG = LoggerFactory.getLogger(SearchServlet.class); 1 usage
16
17     @Override 1 usage new *
18     protected void doGet(SlingHttpServletRequest request, SlingHttpServletResponse response) throws ServletException,
19         String searchText = request.getParameter("query"); if (searchText == null || searchText.isEmpty()) {
20             response.getWriter().write("Query parameter is required."); return;
21         }
22
23         try {
24             QueryBuilder queryBuilder = request.getResourceResolver().adaptTo(QueryBuilder.class); Session session =
25
26             Map<String, String> queryMap = new HashMap<>(); queryMap.put("path", "/content/us/en"); queryMap.put("typ
27
28             Query query = queryBuilder.createQuery(queryBuilder.createQuery(queryMap), session); SearchResult result
29
30             List<Hit> hits = result.getHits();
31
32             response.setContentType("application/json"); response.getWriter().write("{\"results\": ["); for (Hit
33                 response.getWriter().write("{\"path\": \"" + hit.getPath() + "\" },");
34             }
35             response.getWriter().write("]");
36         } catch (Exception e) { LOG.error("Search error: ", e);
37             response.getWriter().write("Error occurred during search.");
38         }
39     }
```