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

```
import org.osgi.service.component.annotations.Component;
                                                                                                      A4 ^
        import org.apache.sling.servlets.annotations.SlingServletResourceTypes; import org.slf4j.Logger;
       import org.slf4j.LoggerFactory;
                                                                                                                  m
       import javax.servlet.Servlet;
       import javax.servlet.ServletException; import java.io.IOException;
       @Component(service = Servlet.class) @SlingServletResourceTypes( new*
               resourceTypes = "assignment/components/sample", methods = {"GET", "POST"},
               extensions = "json"
       public class SampleServlet extends SlingAllMethodsServlet {
           private static final Logger LOG = LoggerFactory.getLogger(SampleServlet.class); 2 usages
@<sup>1</sup>@
           protected void doGet(SlingHttpServletRequest request, SlingHttpServletResponse response) throws Serv
               LOG.info("Sample Servlet GET Request"); response.setContentType("application/json"); response.ge
           protected void doPost(SlingHttpServletRequest request, SlingHttpServletResponse response) throws Ser
@ @
               LOG.info("Sample Servlet POST Request"); response.setContentType("application/json"); response.g
```

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

```
import org.slf4j.LoggerFactory;
                                                                                                    ∆3 ×1 ^ ∨
                                                                                                                      ම
                                                                                                                      m
@Component(service = Servlet.class) @SlingServletPaths("/bin/createPage") new*
public class CreatePageServlet extends SlingSafeMethodsServlet {
   private static final Logger LOG = LoggerFactory.getLogger(CreatePageServlet.class); 3 usages
    protected void doGet(SlingHttpServletRequest request, SlingHttpServletResponse response) throws ServletException,
       String pageName = request.getParameter( s: "pageName"); if (pageName == null || pageName.isEmpty()) {
           response.getWriter().write( s: "Page name is required."); return;
           PageManager pageManager = request.getResourceResolver().adaptTo(PageManager.class);
           if (pageManager != null) {
               Page newPage = pageManager.create( s: "/content/us/en", pageName, s2: "/conf/wknd/settings/wcm/templa
               LOG.info("Page created successfully at: {}", newPage.getPath()); response.getWriter().write( s: "Page
           } else {
               LOG.error("PageManager is null"); response.getWriter().write( s: "Error: PageManager is null.");
           response.getWriter().write( s: "Error creating page.");
```

- 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 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

```
© CreatePageServlet.java
                                                                              © SearchServlet.java ×
                                                                                                    01 A3 ^
import javax.servlet.ServletException; import java.io.IOException;
                                                                                                                       m
public class SearchServlet extends SlingSafeMethodsServlet {
   private static final Logger LOG = LoggerFactory.getLogger(SearchServlet.class); 1usage
   protected void doGet(SlingHttpServletRequest request, SlingHttpServletResponse response) throws ServletException,
        String searchText = request.getParameter( s: "query"); if (searchText == null || searchText.isEmpty()) {
            response.getWriter().write( s: "Query parameter is required."); return;
        try {
            {\tt QueryBuilder\ queryBuilder\ =\ request.getResourceResolver().adaptTo(QueryBuilder.class);\ Session\ session}
            Map<String, String> queryMap = new HashMap<>(); queryMap.put("path", "/content/us/en"); queryMap.put("typ
            Query query = queryBuilder.createQuery(queryBuilder.createQuery(queryMap), session); SearchResult result
            List<Hit> hits = result.getHits();
            response.setContentType("application/json"); response.getWriter().write( s: "{ \"results\": ["); for (Hit
               response.getWriter().write( s: "{ \"path\": \"" + hit.getPath() + "\" },");
            response.getWriter().write( s: "] }");
            response.getWriter().write( s: "Error occurred during search.");
```