AEM Training - Day 7 Assignment (26-03-2025)

1. Create `SampleServlet` Extending `SlingAllMethodsServlet`

Description:

AEM Servlet registered using `resourceType`.
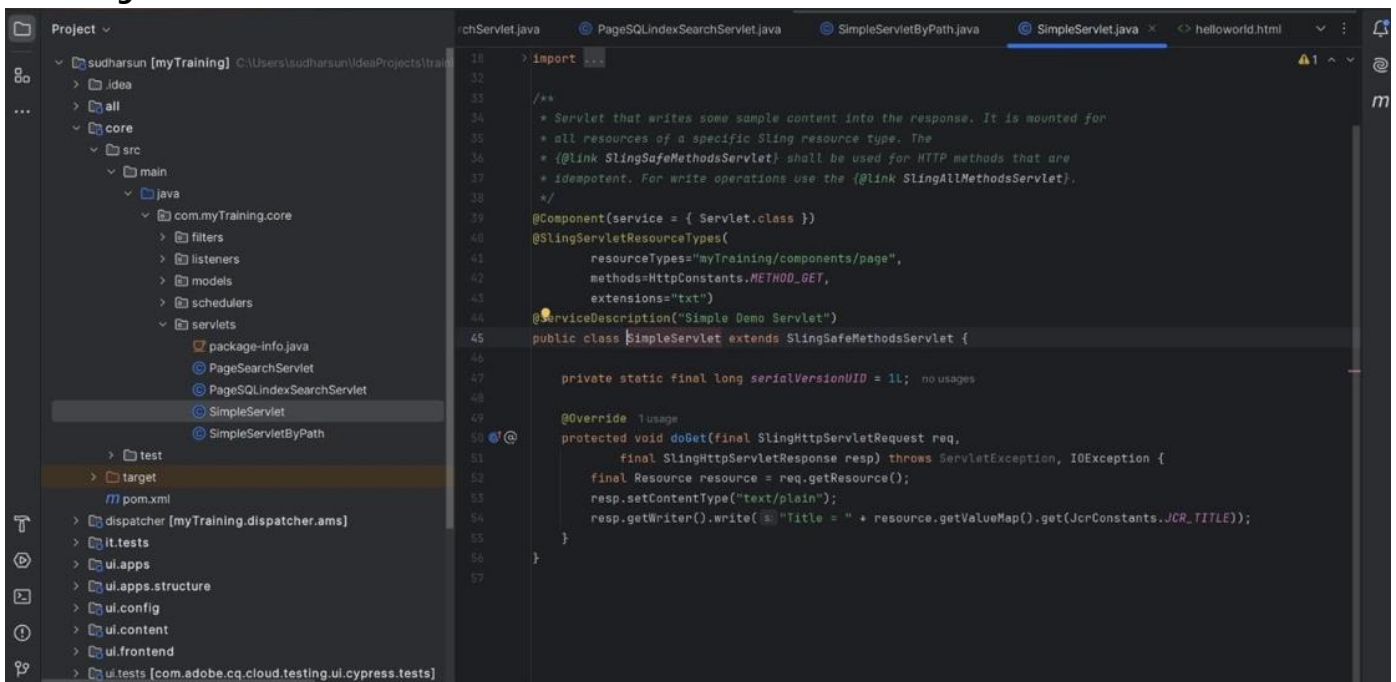
Code Snippet (`SampleServlet.java`):

```java
@Component(service = Servlet.class, property = {
    "sling.servlet.resourceTypes=myapp/components/page",
    "sling.servlet.methods=GET"
})
public class SampleServlet extends SlingAllMethodsServlet {
    private static final Logger LOG =
LoggerFactory.getLogger(SampleServlet.class);

    @Override
    protected void doGet(SlingHttpServletRequest request, SlingHttpServletResponse
response) throws ServletException, IOException {
        response.getWriter().write("Sample Servlet Executed");
        LOG.info("Sample Servlet Executed");
    }
}
```

Screenshots

1. **SlingAllMethodsServlet**

2. Create `CreatePageServlet` Extending `SlingSafeMethodsServlet`
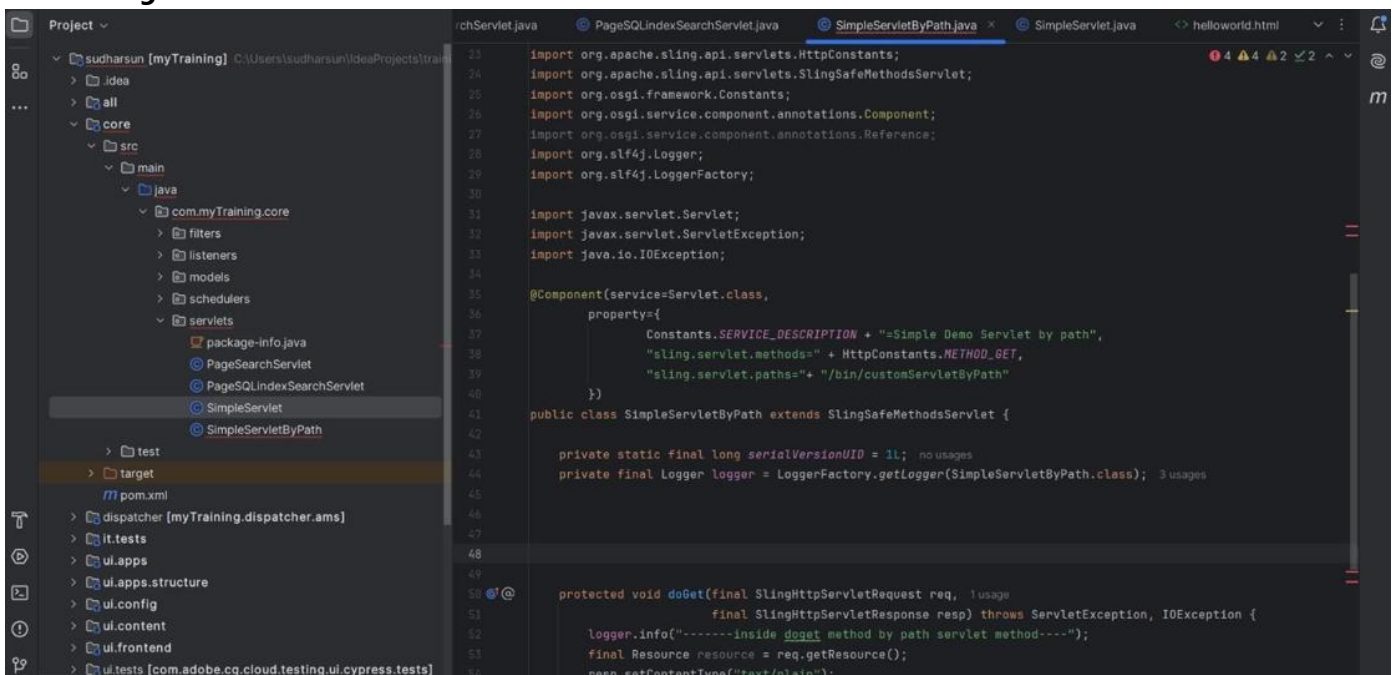
Description:

A Servlet registered using **path** that creates pages based on user input.

Code Snippet (`CreatePageServlet.java`):

```java
@Component(service = Servlet.class, property = {
    "sling.servlet.paths=/bin/createPage",
"sling.servlet.methods=POST"
}) public class CreatePageServlet extends
SlingSafeMethodsServlet {
    @Reference      private ResourceResolverFactory resolverFactory;

    @Override
    protected void doPost(SlingHttpServletRequest request,
SlingHttpServletResponse response) throws ServletException, IOException {
String pageName = request.getParameter("pageName");        try
(ResourceResolver resolver = resolverFactory.getServiceResourceResolver(null))
{
        PageManager pageManager = resolver.adaptTo(PageManager.class);
if (pageManager != null) {
            Page newPage = pageManager.create("/content/mysite", pageName,
"/apps/myapp/templates/page", pageName);
resolver.commit();              response.getWriter().write("Page
created successfully: " + newPage.getPath());            } else {
response.getWriter().write( "PageManager is null" );              }
    } catch (PersistenceException e) {
        response.getWriter().write( "Page creation failed: " + e.getMessage());
}
    }
}
```

1. **SlingSafeMethodsServlet**

3.   Use PageManager API to Create Pages

Description:

The above servlet uses the **PageManager API** to create pages dynamically.

Steps:

1. Get the PageManager instance from ResourceResolver.
2. Call the create() method with parameters: parent path, page name, template path, and title.
3. Commit the changes using resolver.commit().

Code Snippet:

```java
PageManager pageManager = resolver.adaptTo(PageManager.class);
if (pageManager != null) {
    Page newPage = pageManager.create("/content/mysite", pageName,
"/apps/myapp/templates/page", pageName);
    resolver.commit();
}
```

4.   Create SearchServlet Using PredicateMap

Description:
AEM Servlet to search content using **QueryBuilder API**.

Code Snippet (SearchServlet.java):

```java
@Component(service = Servlet.class, property = {
    "sling.servlet.paths=/bin/searchContent",
    "sling.servlet.methods=GET"
})
public class SearchServlet extends SlingSafeMethodsServlet {
    @Reference
    private QueryBuilder queryBuilder;

    @Override
    protected void doGet(SlingHttpServletRequest request, SlingHttpServletResponse
response) throws ServletException, IOException {
        Map<String, String> map = new HashMap<>();
        map.put("path", "/content/mysite");
        map.put("type", "cq:Page");
        map.put("p.limit", "-1");

        Query query = queryBuilder.createQuery(PredicateGroup.create(map),
request.getResourceResolver().adaptTo(Session.class));
        SearchResult result = query.getResult();
        for (Hit hit : result.getHits()) {
            response.getWriter().write(hit.getPath() + "\n");
        }
    }
}
```