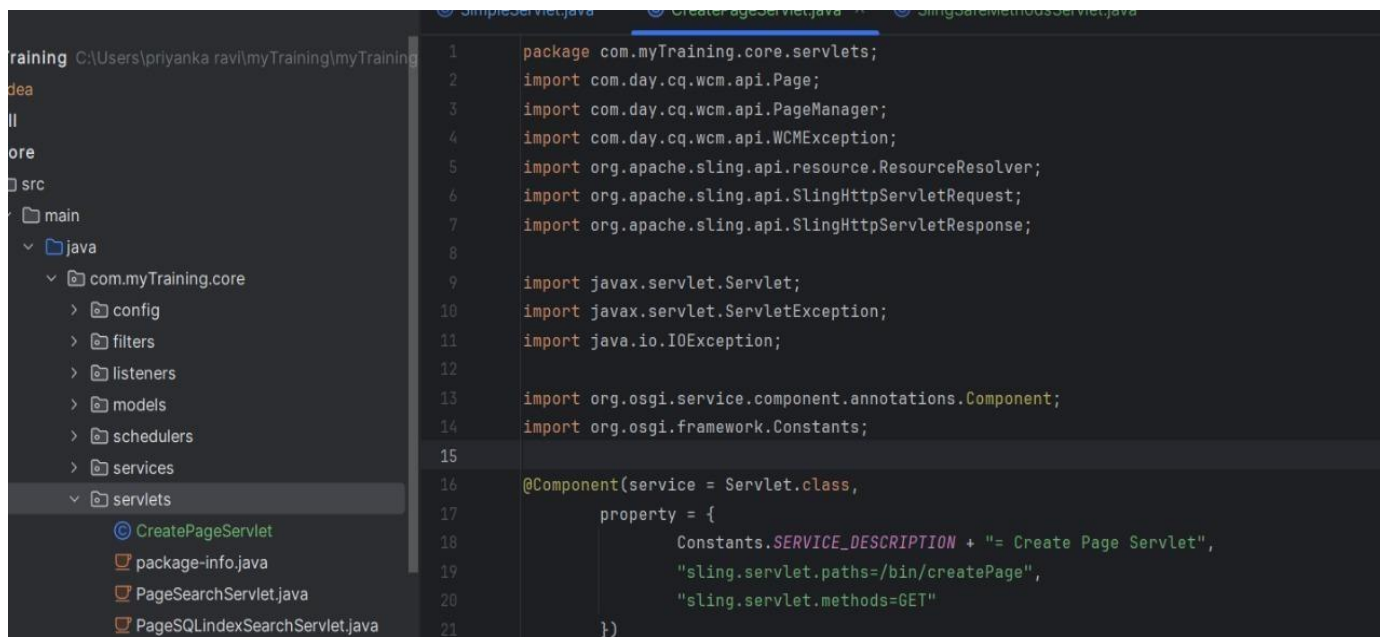


# AEM Task – 7: AEM Servlets and Page Management

## Created SampleServlet (Using resourceType)

- Developed a servlet using **SlingAllMethodsServlet** and registered it with **resourceType**.
- The servlet was designed to handle both **GET** and **POST** requests.
- It processed user inputs and returned appropriate responses based on the requested resource.
- Deployment was completed, and the servlet's registration was verified in the **AEM Web Console**.



```
1 package com.myTraining.core.servlets;
2 import com.day.cq.wcm.api.Page;
3 import com.day.cq.wcm.api.PageManager;
4 import com.day.cq.wcm.api.WCMException;
5 import org.apache.sling.api.resource.ResourceResolver;
6 import org.apache.sling.api.SlingHttpServletRequest;
7 import org.apache.sling.api.SlingHttpServletResponse;
8
9 import javax.servlet.ServletException;
10 import javax.servlet.ServletException;
11 import java.io.IOException;
12
13 import org.osgi.service.component.annotations.Component;
14 import org.osgi.framework.Constants;
15
16 @Component(service = Servlet.class,
17     property = {
18         Constants.SERVICE_DESCRIPTION + "= Create Page Servlet",
19         "sling.servlet.paths=/bin/createPage",
20         "sling.servlet.methods=GET"
21     })
```

## Created CreatePageServlet (Using Path)

- Developed a servlet using **SlingSafeMethodsServlet** and registered it with a specific path (**/bin/createPage**).
- It handled requests to dynamically create pages in AEM based on user input.
- The servlet received the page name, validated its uniqueness, and created the page within a predefined content structure.
- The **PageManager API** was utilized to perform the page creation.
- The servlet was successfully deployed, tested, and verified.

```
1 > /.../  
16 package com.myTraining.core.servlets;  
17 import org.apache.sling.api.SlingHttpServletRequest;  
18 import org.apache.sling.api.SlingHttpServletResponse;  
19  
20 import javax.servlet.Servlet;  
21 import javax.servlet.ServletException;  
22 import java.io.IOException;  
23  
24 import org.osgi.service.component.annotations.Component;  
25 import org.osgi.framework.Constants;  
26  
27 @Component(service = Servlet.class,  
28             property = {  
29                 Constants.SERVICE_DESCRIPTION + "= Simple Servlet using Resource Type",  
30                 "sling.servlet.methods=GET",  
31                 "sling.servlet.methods=POST",  
32                 "sling.servlet.resourceTypes=example/components/page",  
33                 "sling.servlet.extensions=json"  
34             })  
35 public class SimpleServlet extends SlingSafeMethodsServlet {  
36     @Override
```

## Accepted User Input for Page Creation

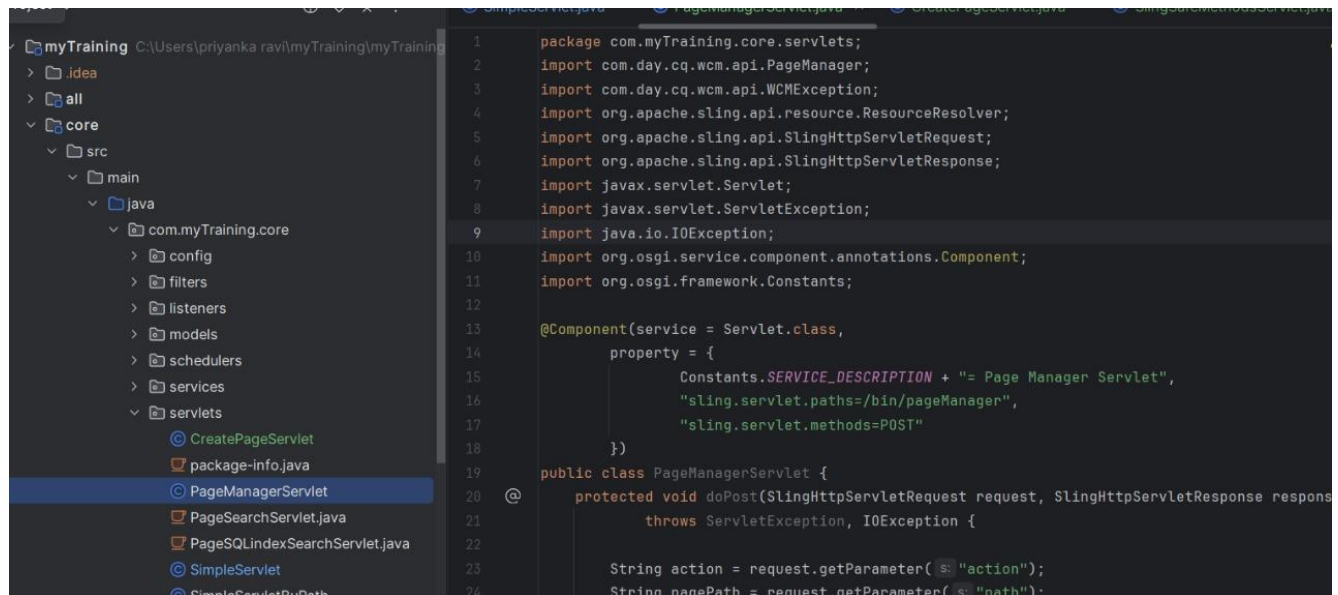
- The servlet was configured to accept a **page name** from the user.
- A validation process was implemented to ensure **duplicate pages** were not created.
- Once validated, the **PageManager API** was used to dynamically create the new page.
- After execution, a **confirmation message** was logged and displayed.

```
1 package com.myTraining.core.servlets;  
2 import com.day.cq.wcm.api.PageManager;  
3 import com.day.cq.wcm.api.WCMException;  
4 import org.apache.sling.api.resource.ResourceResolver;  
5 import org.apache.sling.api.SlingHttpServletRequest;  
6 import org.apache.sling.api.SlingHttpServletResponse;  
7 import javax.servlet.Servlet;  
8 import javax.servlet.ServletException;  
9 import java.io.IOException;  
10 import org.osgi.service.component.annotations.Component;  
11 import org.osgi.framework.Constants;  
12  
13 @Component(service = Servlet.class,  
14             property = {  
15                 Constants.SERVICE_DESCRIPTION + "= Page Manager Servlet",  
16                 "sling.servlet.paths=/bin/pageManager",  
17                 "sling.servlet.methods=POST"  
18             })  
19 public class PageManagerServlet {  
20     @Override  
21     protected void doPost(SlingHttpServletRequest request, SlingHttpServletResponse response)  
22         throws ServletException, IOException {  
23  
24         String action = request.getParameter("action");  
25         String pagePath = request.getParameter("path");
```

## Managed Pages Using PageManager API

- The **PageManager API** was used to create, retrieve, modify, and delete AEM pages.
- Appropriate permissions were configured to enable the servlet to perform these operations

Functional testing was carried out to verify the successful execution of page-related tasks.



## Created SearchServlet for Content Search

- Developed and registered a servlet at **/bin/search** to enable content search functionality.
- Integrated the **Query Builder API** using **PredicateMap** to execute structured queries.
- Configured the search functionality with parameters such as:
  - **Path filtering** to restrict searches to **/content/my-site**.
  - **Page type filtering** to search only within **cq:Page** nodes.
  - **Full-text search** to retrieve relevant content based on user input.
- The servlet was deployed, tested, and verified to return accurate search results.



