

# Task – 7

## Created SampleServlet (Using resourceType)

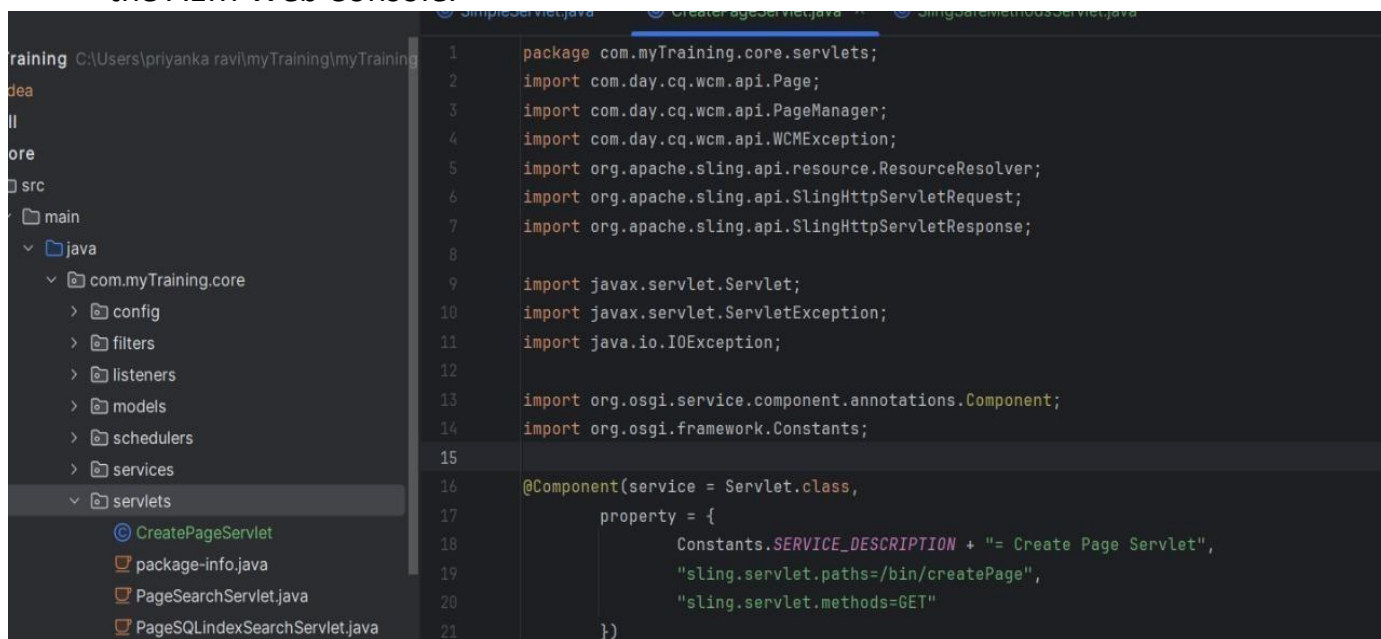
- A custom servlet named SampleServlet was developed using SlingAllMethodsServlet to handle multiple HTTP methods.

### Key Implementation Details:

- Servlet Registration:
  - Registered using the `@SlingServletResourceTypes` annotation.
  - Mapped to a specific resourceType, making it accessible via components or content using that type.
- HTTP Method Handling:
  - GET: Used to fetch and return information based on the resource.
  - POST: Used to process user-submitted data and respond accordingly.
- Functionality:
  - Processes user inputs (e.g., form data or request parameters).
  - Returns custom responses depending on the logic and resource context.

This servlet enhances flexibility by allowing business logic to be tightly coupled with component types, rather than static paths or selectors.

- 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.Servlet;
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)

- A custom servlet named CreatePageServlet was implemented using SlingSafeMethodsServlet, tailored for safe operations like content reading and non-

destructive actions.

#### Key Highlights:

- Servlet Registration:
  - Mapped to the path `/bin/createPage` using the `@SlingServletPaths` annotation.
  - Makes the servlet accessible via a direct HTTP request to this endpoint.
- Functionality:
  - Accepts user input such as the desired page name.
  - Validates whether the page name already exists to avoid duplication.
  - Creates a new page under a specified parent path using predefined templates and content structure.
- Uses the PageManager API:
  - Facilitates the creation of pages programmatically in the JCR repository.
  - Handles setting up the necessary properties and structure for the new page.
- Deployment and Testing:
  - The servlet was successfully deployed to the AEM instance.
  - Tested and verified to ensure it performs correctly across valid and invalid input scenarios.

This servlet enhances content automation and can be extended for bulk page creation, integration with external systems, or custom authoring tools.

```

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.ServletException;
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.

```

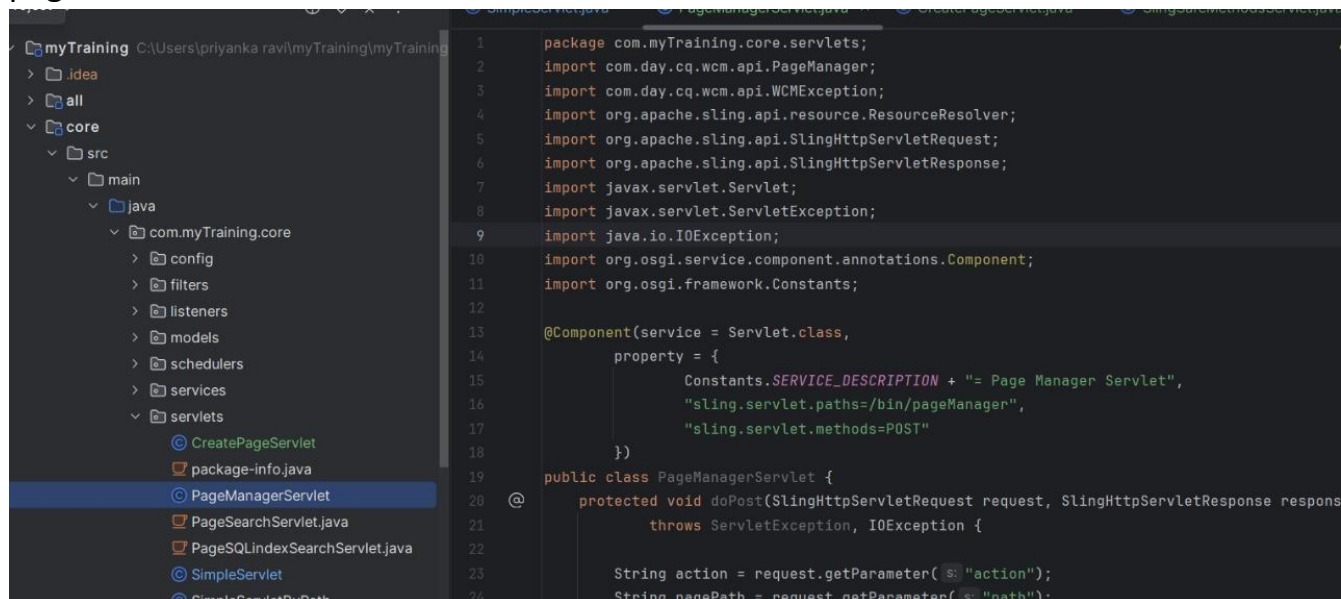
1  package com.myTraining.core.servlets;
2  import com.day.cq.wcm.api.PageManager;
3  import com.day.cq.wcm.api.WCMException;
4  import org.apache.sling.api.resource.ResourceResolver;
5  import org.apache.sling.api.SlingHttpServletRequest;
6  import org.apache.sling.api.SlingHttpServletResponse;
7  import javax.servlet.ServletException;
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     @protected void doPost(SlingHttpServletRequest request, SlingHttpServletResponse response)
21     throws ServletException, IOException {
22
23         String action = request.getParameter("action");
24         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.



