Digital Asset Management (DAM) and Tasks Execution Guide

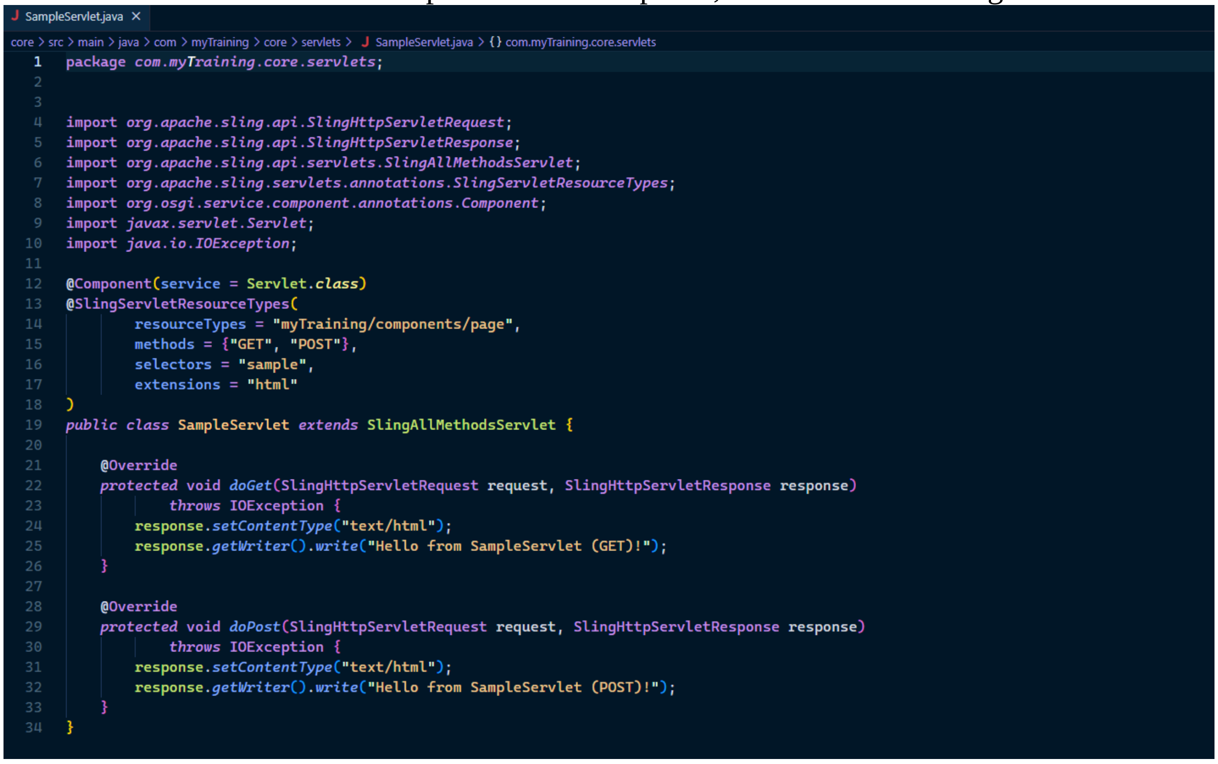
---

1. SampleServlet (resourceType-based registration) Check Output:

- **URL Format:** http://localhost:4502/content/samplepage.sample.json •

Replace samplepage with the actual page where you registered the servlet.

- If it returns the expected JSON response, the servlet is working.

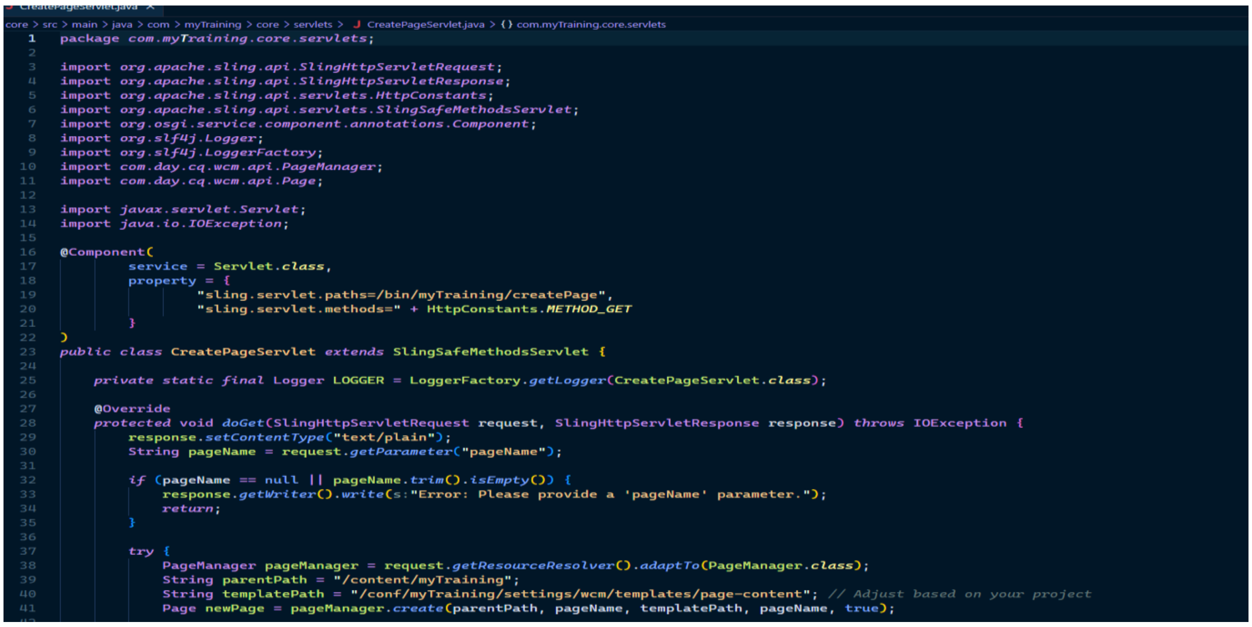


SampleServlet.java

core > src > main > java > com > myTraining > core > servlets > SampleServlet.java > {} com.myTraining.core.servlets

```java
package com.myTraining.core.servlets;


import org.apache.sling.api.SlingHttpServletRequest;
import org.apache.sling.api.SlingHttpServletResponse;
import org.apache.sling.api.servlets.SlingAllMethodsServlet;
import org.apache.sling.servlets.annotations.SlingServletResourceTypes;
import org.osgi.service.component.annotations.Component;
import javax.servlet.Servlet;
import java.io.IOException;

@Component(service = Servlet.class)
@SlingServletResourceTypes(
        resourceTypes = "myTraining/components/page",
        methods = {"GET", "POST"},
        selectors = "sample",
        extensions = "html"
)
public class SampleServlet extends SlingAllMethodsServlet {

    @Override
    protected void doGet(SlingHttpServletRequest request, SlingHttpServletResponse response)
            throws IOException {
        response.setContentType("text/html");
        response.getWriter().write("Hello from SampleServlet (GET)!");
    }

    @Override
    protected void doPost(SlingHttpServletRequest request, SlingHttpServletResponse response)
            throws IOException {
        response.setContentType("text/html");
        response.getWriter().write("Hello from SampleServlet (POST)!");
    }
}
```

2. CreatePageServlet (Registered via path)

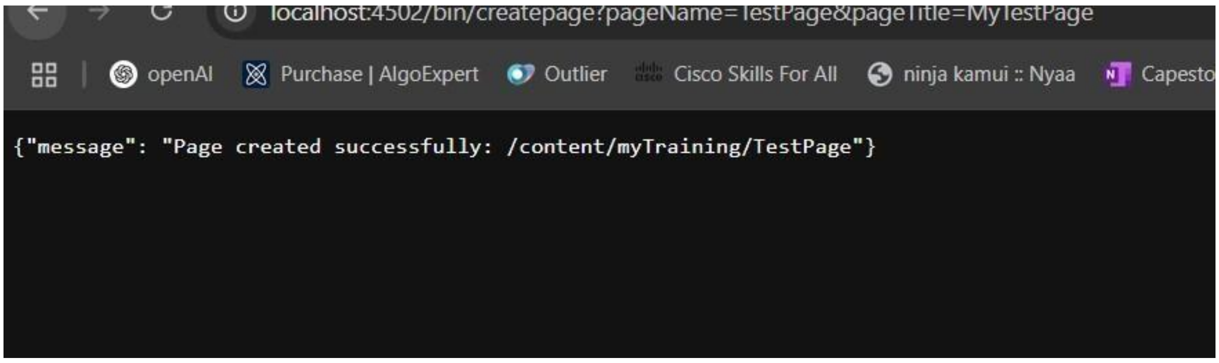


CreatePageServlet.java

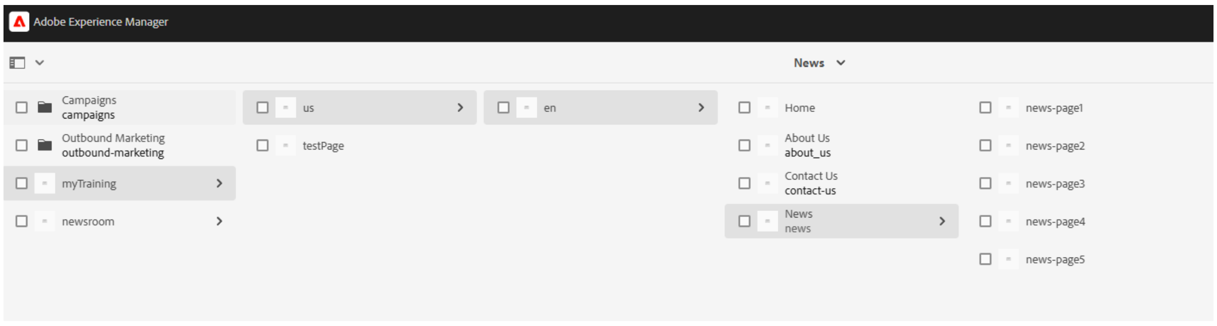
core > src > main > java > com > myTraining > core > servlets > CreatePageServlet.java > {} com.myTraining.core.servlets

```java
package com.myTraining.core.servlets;

import org.apache.sling.api.SlingHttpServletRequest;
import org.apache.sling.api.SlingHttpServletResponse;
import org.apache.sling.api.servlets.HttpConstants;
import org.apache.sling.api.servlets.SlingSafeMethodsServlet;
import org.osgi.service.component.annotations.Component;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import com.day.cq.wcm.api.PageManager;
import com.day.cq.wcm.api.Page;

import javax.servlet.Servlet;
import java.io.IOException;

@Component(
        service = Servlet.class,
        property = {
                "sling.servlet.paths=/bin/myTraining/createPage",
                "sling.servlet.methods=" + HttpConstants.METHOD_GET
        }
)
public class CreatePageServlet extends SlingSafeMethodsServlet {

    private static final Logger LOGGER = LoggerFactory.getLogger(CreatePageServlet.class);

    @Override
    protected void doGet(SlingHttpServletRequest request, SlingHttpServletResponse response) throws IOException {
        response.setContentType("text/plain");
        String pageName = request.getParameter("pageName");

        if (pageName == null || pageName.trim().isEmpty()) {
            response.getWriter().write(s:"Error: Please provide a 'pageName' parameter.");
            return;
        }

        try {
            PageManager pageManager = request.getResourceResolver().adaptTo(PageManager.class);
            String parentPath = "/content/myTraining";
            String templatePath = "/conf/myTraining/settings/wcm/templates/page-content"; // Adjust based on your project
            Page newPage = pageManager.create(parentPath, pageName, templatePath, pageName, true);
```

Check Output:



```
{"message": "Page created successfully: /content/myTraining/TestPage"}
```



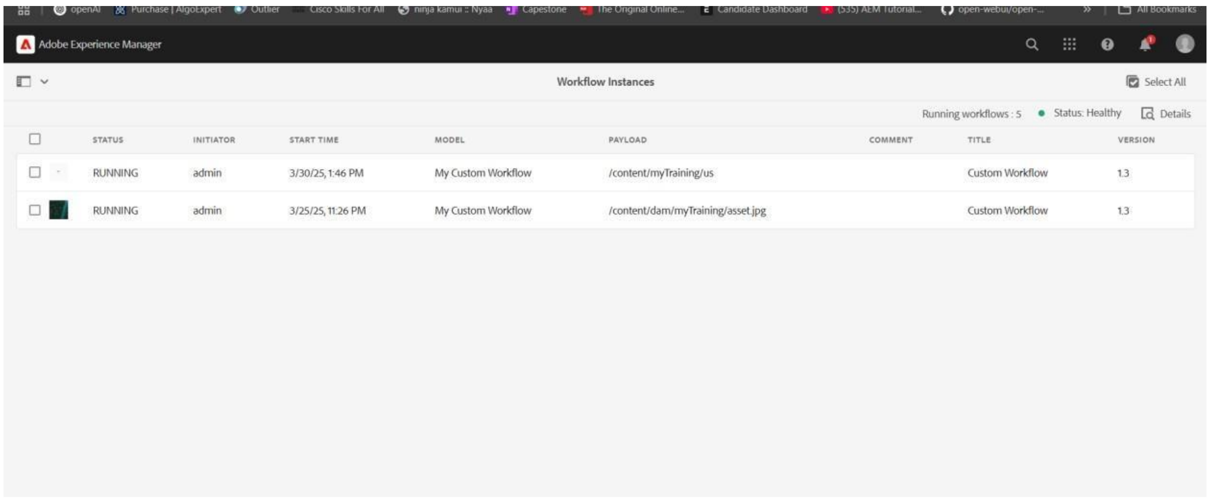
3.Page Creation Using PageManager API

Check Output:

- **Steps:**

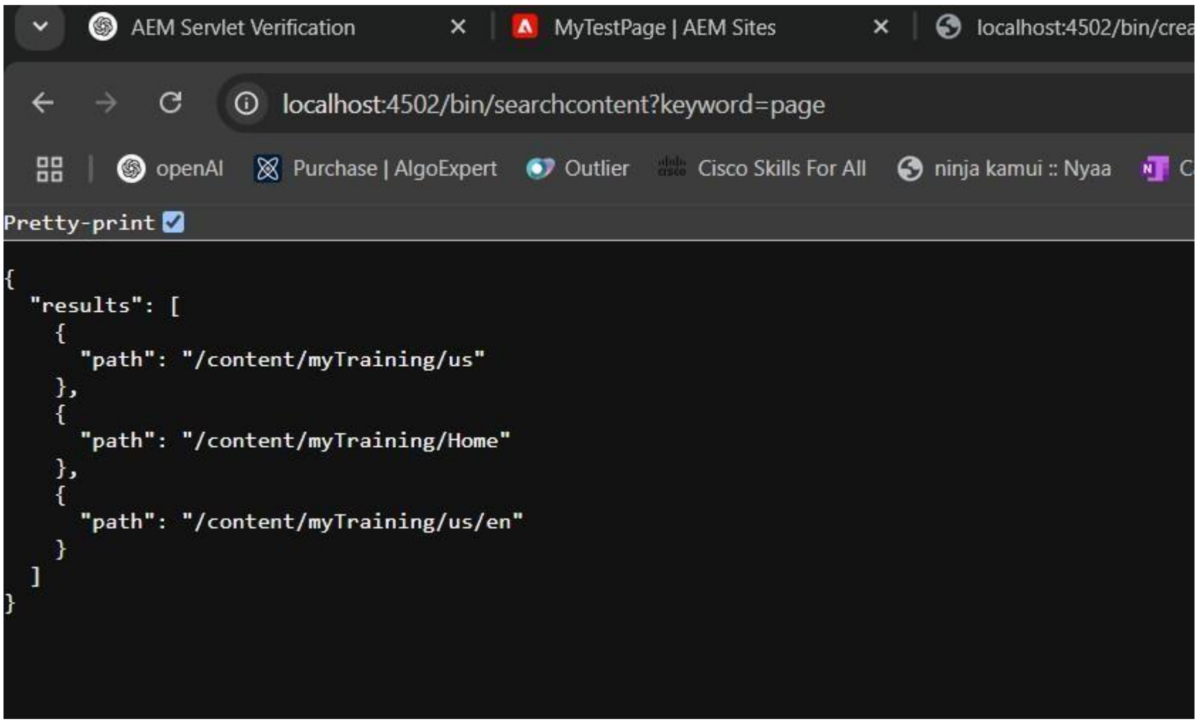
1. Run CreatePageServlet (Step 2).

Page created:



STATUS	INITIATOR	START TIME	MODEL	PAYLOAD	COMMENT	TITLE	VERSION
RUNNING	admin	3/30/25, 1:46 PM	My Custom Workflow	/content/myTraining/us		Custom Workflow	1.3
RUNNING	admin	3/25/25, 11:26 PM	My Custom Workflow	/content/dam/myTraining/asset.jpg		Custom Workflow	1.3

2. SearchServlet (PredicateMap-based search)



```
{
  "results": [
    {
      "path": "/content/myTraining/us"
    },
    {
      "path": "/content/myTraining/Home"
    },
    {
      "path": "/content/myTraining/us/en"
    }
  ]
}
```