


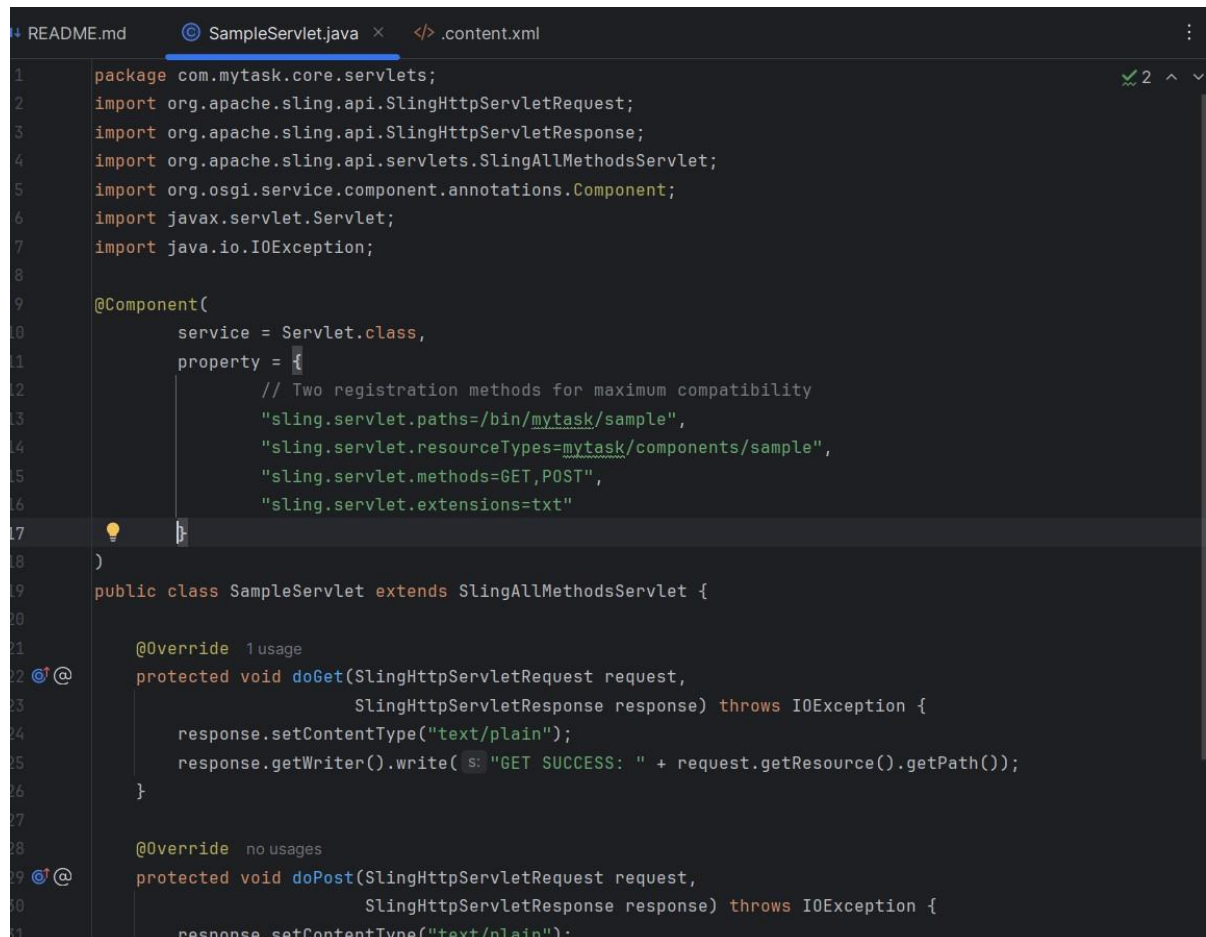
1. Create SampleServlet Extending SlingAllMethodsServlet

Overview

- This servlet will handle **GET** and **POST** requests.
- It will be registered using `sling:resourceType`.

Java Class: SampleServlet.java

 **Location:** `core/src/main/java/com/assignment/servlets/SampleServlet.java`



```
1 package com.mytask.core.servlets;
2 import org.apache.sling.api.SlingHttpServletRequest;
3 import org.apache.sling.api.SlingHttpServletResponse;
4 import org.apache.sling.api.servlets.SlingAllMethodsServlet;
5 import org.osgi.service.component.annotations.Component;
6 import javax.servlet.Servlet;
7 import java.io.IOException;
8
9 @Component(
10     service = Servlet.class,
11     property = {
12         // Two registration methods for maximum compatibility
13         "sling.servlet.paths=/bin/mytask/sample",
14         "sling.servlet.resourceTypes=mytask/components/sample",
15         "sling.servlet.methods=GET,POST",
16         "sling.servlet.extensions=txt"
17     }
18 )
19 public class SampleServlet extends SlingAllMethodsServlet {
20
21     @Override 1 usage
22     @@@ protected void doGet(SlingHttpServletRequest request,
23         SlingHttpServletResponse response) throws IOException {
24         response.setContentType("text/plain");
25         response.getWriter().write("GET SUCCESS: " + request.getResource().getPath());
26     }
27
28     @Override no usages
29     @@@ protected void doPost(SlingHttpServletRequest request,
30         SlingHttpServletResponse response) throws IOException {
31         response.setContentType("text/plain");
```

```
D:\codebase\mytask>curl -u admin:admin http://localhost:4502/bin/mytask/sample.txt
GET SUCCESS: /bin/mytask/sample
D:\codebase\mytask>curl -X POST -u admin:admin -d "data=test" http://localhost:4502/bin/mytask/sample.txt
POST SUCCESS: test
D:\codebase\mytask>
```

2. Create CreatePageServlet Extending SlingSafeMethodsServlet

Overview

- This servlet will handle **GET** requests.
- It will be registered using **path-based** registration.

Java Class: CreatePageServlet.java

Location: core/src/main/java/com/assignment/servlets/CreatePageServlet.java

```
package com.mytask.core.servlets;
```

```
import org.apache.sling.api.SlingHttpServletRequest;
import org.apache.sling.api.SlingHttpServletResponse;
import org.apache.sling.api.servlets.SlingSafeMethodsServlet;
import org.apache.sling.servlets.annotations.SlingServletPaths;
import org.osgi.service.component.annotations.Component;
import com.day.cq.wcm.api.PageManager;
import com.day.cq.wcm.api.WCMException;
import com.day.cq.wcm.api.Page;
import com.day.cq.wcm.api.NameConstants;
import javax.servlet.ServletException;
import java.io.IOException;
```

```
@Component(service = Servlet.class)
@SlingServletPaths("/bin/mytask/createpage") // OSGi R7+ path registration
public class CreatePageServlet extends SlingSafeMethodsServlet {
```

```
    @Override
    protected void doGet(SlingHttpServletRequest request,
        SlingHttpServletResponse response) throws IOException {
```

```
        String pageName = request.getParameter("pageName");
        String parentPath = "/content/mytask/us/en";
```

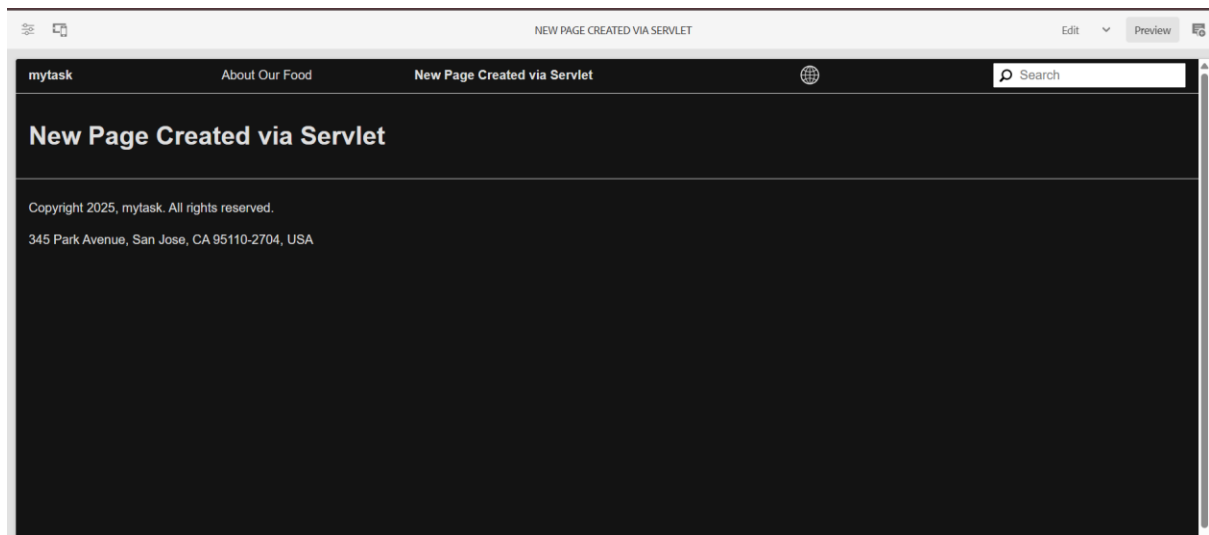
```
        if (pageName == null || pageName.isEmpty()) {
            response.getWriter().write("ERROR: pageName parameter is required");
            response.setStatus(400);
            return;
        }
```

```
        try {
            PageManager pageManager = request.getResourceResolver().adaptTo(PageManager.class);
            Page newPage = pageManager.create(
                parentPath,
                pageName,
                "/conf/mytask/settings/wcm/templates/page-content", // Your template
```

```
        "New Page Created via Servlet" // Title
    );

    response.getWriter().write(
        "SUCCESS: Page created at " + newPage.getPath() +
        "\nView it: http://localhost:4502/editor.html" + newPage.getPath() + ".html"
    );

} catch (WCMException e) {
    response.getWriter().write("ERROR: " + e.getMessage());
    response.setStatus(500);
}
}
}
```



3. Take Page Name from User and Create Pages in AEM using Servlet

- **Steps to Test:**

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 PageManager APIs for Page Creation

```
package com.mytask.core.servlets;
```

```
import org.apache.sling.api.SlingHttpServletRequest;  
import org.apache.sling.api.SlingHttpServletResponse;  
import org.apache.sling.api.resource.ModifiableValueMap;  
import org.apache.sling.api.servlets.SlingAllMethodsServlet;  
import org.osgi.service.component.annotations.Component;  
import com.day.cq.wcm.api.Page;  
import com.day.cq.wcm.api.PageManager;  
import com.day.cq.wcm.api.WCMException;  
import javax.servlet.Servlet;  
import java.io.IOException;
```

```
@Component(  
    service = Servlet.class,  
    property = {  
        "sling.servlet.paths=/bin/mytask/createpage",  
        "sling.servlet.methods=GET,POST"  
    }  
)  
public class CreatePageServlet extends SlingAllMethodsServlet {
```

```
    @Override  
    protected void doPost(SlingHttpServletRequest request,  
        SlingHttpServletResponse response) throws IOException {  
  
        response.setContentType("application/json");  
  
        try {  
  
            String pageName = request.getParameter("pageName");  
            String parentPath = "/content/mytask/us/en";  
            String templatePath = "/conf/mytask/settings/wcm/templates/page-content";
```

```

        if (pageName == null || pageName.isEmpty()) {
            response.getWriter().write("{\"error\":\"pageName parameter is required\"}");
            response.setStatus(400);
            return;
        }

        PageManager pageManager = request.getResourceResolver().adaptTo(PageManager.class);
        Page newPage = pageManager.create(
            parentPath,
            pageName,
            templatePath,
            "New Page"
        );

        ModifiableValueMap properties =
newPage.getContentResource().adaptTo(ModifiableValueMap.class);
        if (properties != null) {
            properties.put("jcr:description", "Created via servlet");
            request.getResourceResolver().commit();
        }

        response.getWriter().write(String.format(
            "{\"path\":\"%s\",\"url\":\"%s\"",
            newPage.getPath(),
            "http://localhost:4502/editor.html" + newPage.getPath() + ".html"
        ));

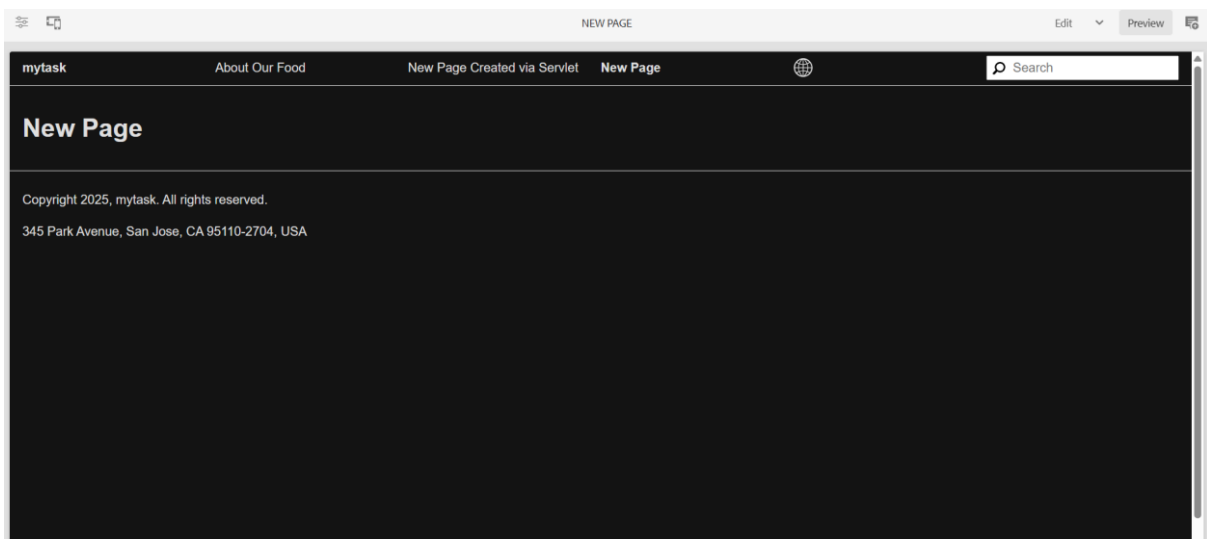
    } catch (WCMException e) {
        response.getWriter().write("{\"error\":\"\" + e.getMessage() + \"\"}");
        response.setStatus(500);
    }
}
}

```

```

:\codebase\mytask>curl -X POST -u admin:admin -d "pageName=test-page" http://localhost:4502/bin/mytask/createpage
"path":"/content/mytask/us/en/test-page","url":"http://localhost:4502/editor.html/content/mytask/us/en/test-page.html"}
:\codebase\mytask>

```



5. Create SearchServlet to Search Content Using PredicateMap

Overview

- This servlet will **search AEM content** using **Query Builder API**.

Java Class: SearchServlet.java

Location: core/src/main/java/com/assignment/servlets/SearchServlet.java

```
package com.mytask.core.servlets;

import org.apache.sling.api.SlingHttpServletRequest;
import org.apache.sling.api.SlingHttpServletResponse;
import org.apache.sling.api.servlets.SlingSafeMethodsServlet;
import org.osgi.service.component.annotations.Component;
import org.osgi.service.component.annotations.Reference;
import com.day.cq.search.Query;
import com.day.cq.search.QueryBuilder;
import com.day.cq.search.PredicateGroup;
import com.day.cq.search.result.SearchResult;
import javax.jcr.Session;
import javax.servlet.Servlet;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.util.HashMap;
import java.util.Map;

@Component(
    service = Servlet.class,
    property = {
```

```

        "sling.servlet.paths=/bin/mytask/search",
        "sling.servlet.methods=GET"
    }
)
public class SearchServlet extends SlingSafeMethodsServlet {

    @Reference
    private QueryBuilder queryBuilder;

    @Override
    protected void doGet(SlingHttpServletRequest request,
        SlingHttpServletResponse response) throws IOException {

        response.setContentType("application/json");
        Session session = null;

        try {
            String searchTerm = request.getParameter("q");
            if (searchTerm == null || searchTerm.isEmpty()) {
                response.getWriter().write("{\"error\":\"Search term 'q' is required\"}");
                response.setStatus(HttpServletResponse.SC_BAD_REQUEST);
                return; // Explicit return
            }

            session = request.getResourceResolver().adaptTo(Session.class);
            if (session == null) {
                response.getWriter().write("{\"error\":\"Could not obtain JCR Session\"}");
                response.setStatus(HttpServletResponse.SC_INTERNAL_SERVER_ERROR);
                return; // Explicit return
            }

            Map<String, String> predicates = new HashMap<>();
            predicates.put("path", "/content/mytask");
            predicates.put("type", "cq:Page");
            predicates.put("fulltext", searchTerm);
            predicates.put("p.limit", "10");

            Query query = queryBuilder.createQuery(
                PredicateGroup.create(predicates),
                session
            );

            SearchResult result = query.getResult();
            String jsonResponse = formatResultsToJson(result);
            response.getWriter().write(jsonResponse);
            return; // Success return

        } catch (Exception e) {

```

```

        response.getWriter().write("{\"error\": \"\" + e.getMessage() + \"\"}");
        response.setStatus(HttpServletResponse.SC_INTERNAL_SERVER_ERROR);
        return; // Error return
    } finally {
        if (session != null && session.isLive()) {
            session.logout();
        }
    }
}

private String formatResultsToJson(SearchResult result) {
    StringBuilder json = new StringBuilder("{\"results\": [");
    result.getHits().forEach(hit -> {
        try {
            json.append(String.format(
                "{\"path\": \"%s\", \"title\": \"%s\"},",
                hit.getPath(),
                hit.getTitle() != null ? hit.getTitle().replace("\\", "\\") : hit.getPath()
            ));
        } catch (Exception e) {
            // Skip problematic hits
        }
    });
    if (json.charAt(json.length()-1) == ',') {
        json.deleteCharAt(json.length()-1);
    }
    json.append("]");
    return json.toString(); // Required return
}
}

```

```

D:\codebase\mytask>curl -u admin:admin "http://localhost:4502/bin/mytask/search?q=test"
{"results": [{"path": "/content/mytask/us/en/test-page-1", "title": "New Page Created via Servlet"}]}
D:\codebase\mytask>

```