

26/03/2025 - TASKS

Table of Contents

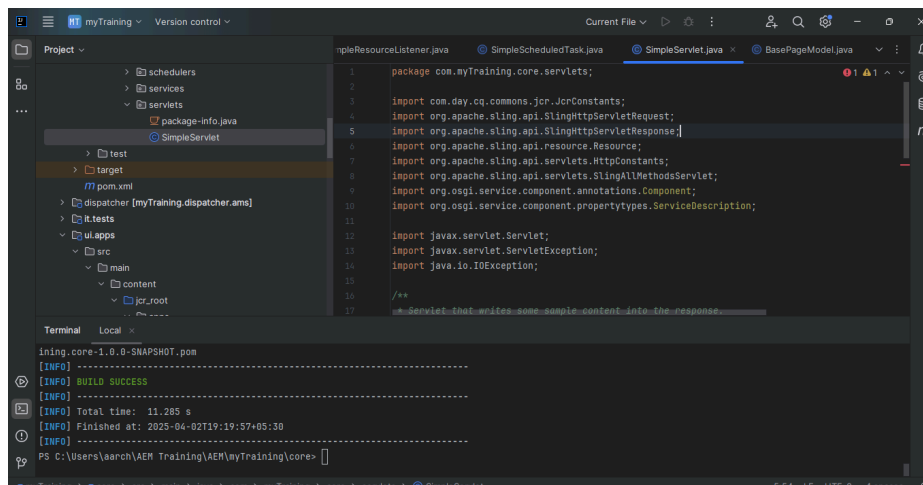
1. Create `SampleServlet`
2. Create `CreatePageServlet`
3. Create Pages in AEM via User Input
4. Implement `SearchServlet` Using `PredicateMap`
5. References

1. Create `SampleServlet`

- Create a servlet that responds with a JSON message.
- Register the servlet using `resourceType`.

Implementation

1. Extend `SlingAllMethodsServlet`.
2. Use `@SlingServletResourceTypes` annotation to register the servlet.
3. Handle `GET` requests and return a JSON response.



The screenshot shows an IDE with the following components:

- Project Explorer:** Shows a project structure with folders like `schedulers`, `services`, `servlets`, `package-info.java`, `test`, `target`, `pom.xml`, `dispatcher [myTraining.dispatcher.ams]`, `it.tests`, `ui.apps`, `src`, `main`, `content`, and `jcr_root`.
- Editor:** Displays the code for `SimpleServlet.java`. The code is as follows:

```
1 package com.myTraining.core.servlets;
2
3 import com.day.cq.commons.jcr.JcrConstants;
4 import org.apache.sling.api.SlingHttpServletRequest;
5 import org.apache.sling.api.SlingHttpServletResponse;
6 import org.apache.sling.api.resource.Resource;
7 import org.apache.sling.api.servlets.HttpConstants;
8 import org.apache.sling.api.servlets.SlingAllMethodsServlet;
9 import org.osgi.service.component.annotations.Component;
10 import org.osgi.service.component.propertytypes.ServiceDescription;
11
12 import javax.servlet.Servlet;
13 import javax.servlet.ServletException;
14 import java.io.IOException;
15
16 /**
17  * A Servlet that writes some sample content into the response.
18  */
```
- Terminal:** Shows the build output for `myTraining-core-1.0.0-SNAPSHOT.pom`. The output indicates a successful build:

```
[INFO] BUILD SUCCESS
[INFO] Total time: 11.285 s
[INFO] Finished at: 2025-04-02T19:19:57:05:30
[INFO] -----
PS C:\Users\laorch\AEM Training\AEM\myTraining\core>
```

```

@Designate(ocd = SampleServlet.Config.class)
@SlingServletResourceTypes(resourceTypes = "myTraining/components/sample", methods =
{"GET"}, extensions = "json")
public class SampleServlet extends SlingAllMethodsServlet {
    @Override
    protected void doGet(SlingHttpServletRequest request, SlingHttpServletResponse response)
throws ServletException, IOException {
        response.setContentType("application/json");
        response.getWriter().write("{\"message\": \"Hello from SampleServlet!\"}");
    }
}

```

2. Create **CreatePageServlet**

Objective

- Create a servlet to generate new pages dynamically in AEM.
- Register the servlet using a **path-based approach**.

Implementation

1. Extend **SlingSafeMethodsServlet**.
2. Use **@SlingServletPaths** to define a unique servlet path.
3. Accept a page name from the user and create a new page in AEM.

```

@SlingServletPaths("/bin/createpage")
public class CreatePageServlet extends SlingSafeMethodsServlet {
    @Reference
    private ResourceResolverFactory resolverFactory;

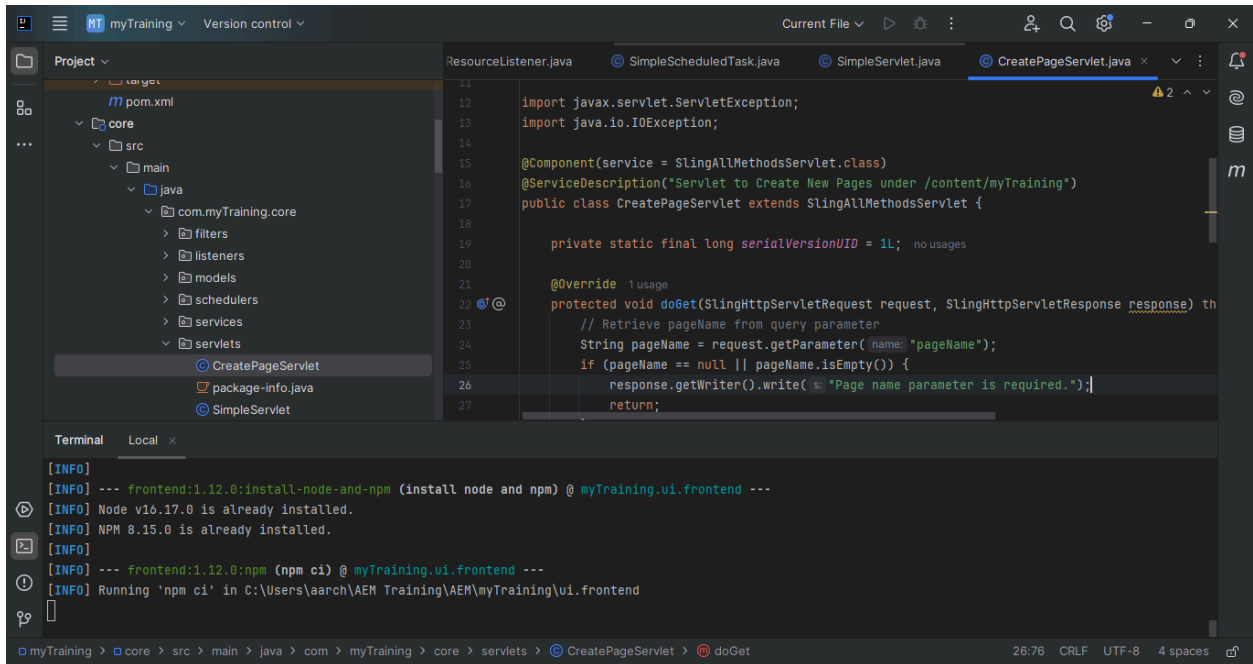
    @Override
    protected void doGet(SlingHttpServletRequest request, SlingHttpServletResponse response)
throws ServletException, IOException {
        String pageName = request.getParameter("pageName");
        try (ResourceResolver resolver = resolverFactory.getServiceResourceResolver(null)) {
            PageManager pageManager = resolver.adaptTo(PageManager.class);
            if (pageManager != null) {

```

```

        Page newPage = pageManager.create("/content/us/en", pageName,
"/apps/wknd/components/page", pageName);
        response.getWriter().write("Page created successfully: " + newPage.getPath());
    }
} catch (Exception e) {
    response.getWriter().write("Error: " + e.getMessage());
}
}
}
}
}

```

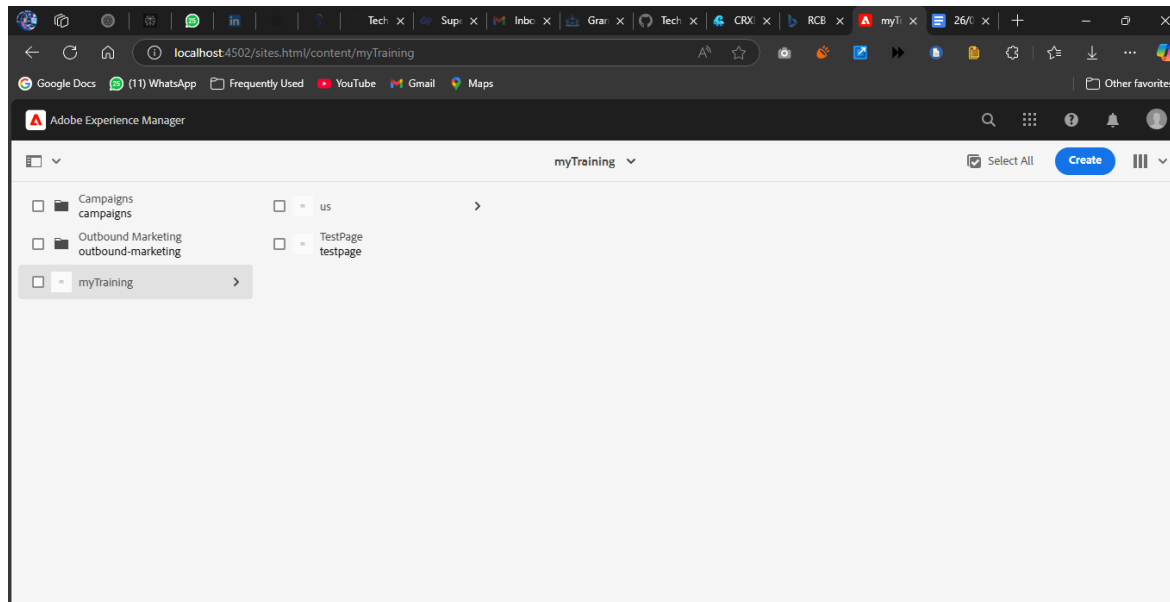


```

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

```

Page Created Successfully:



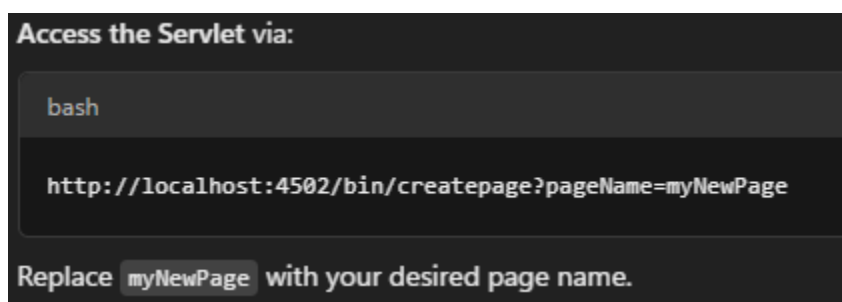
3. Create Pages in AEM via User Input

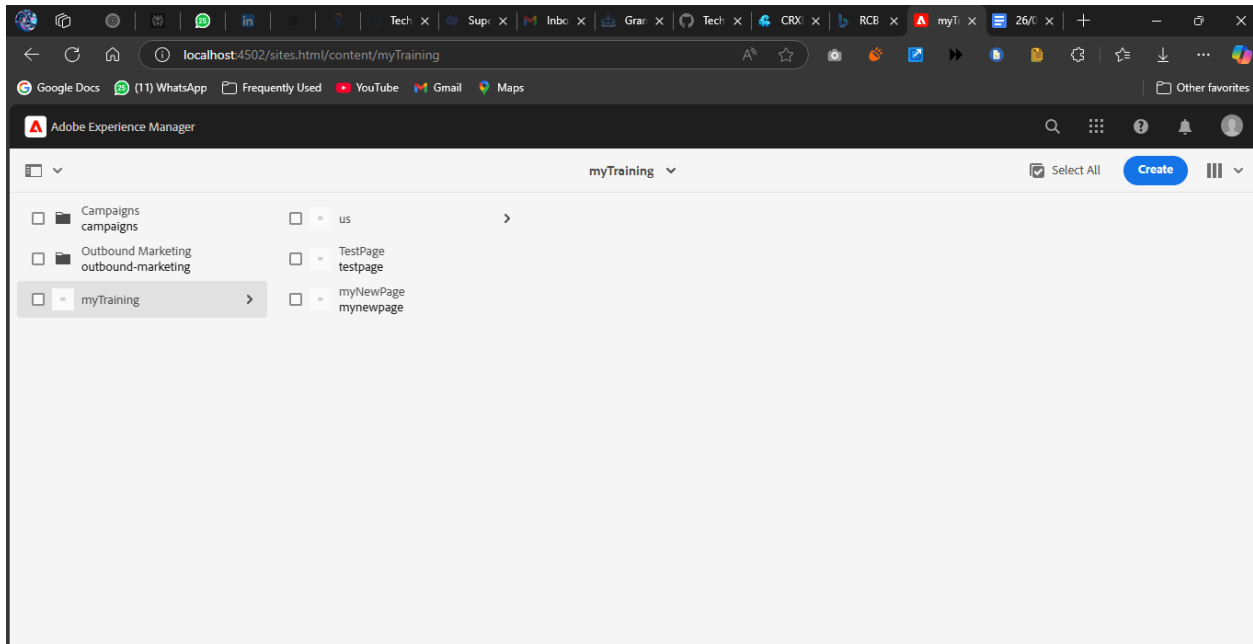
Objective

- Allow users to specify a page name and dynamically generate pages.

Steps

1. The servlet reads the `pageName` from the request parameter.
2. Uses the `PageManager` API to create a new page inside `/content/us/en`.
3. Returns success or error messages as responses.





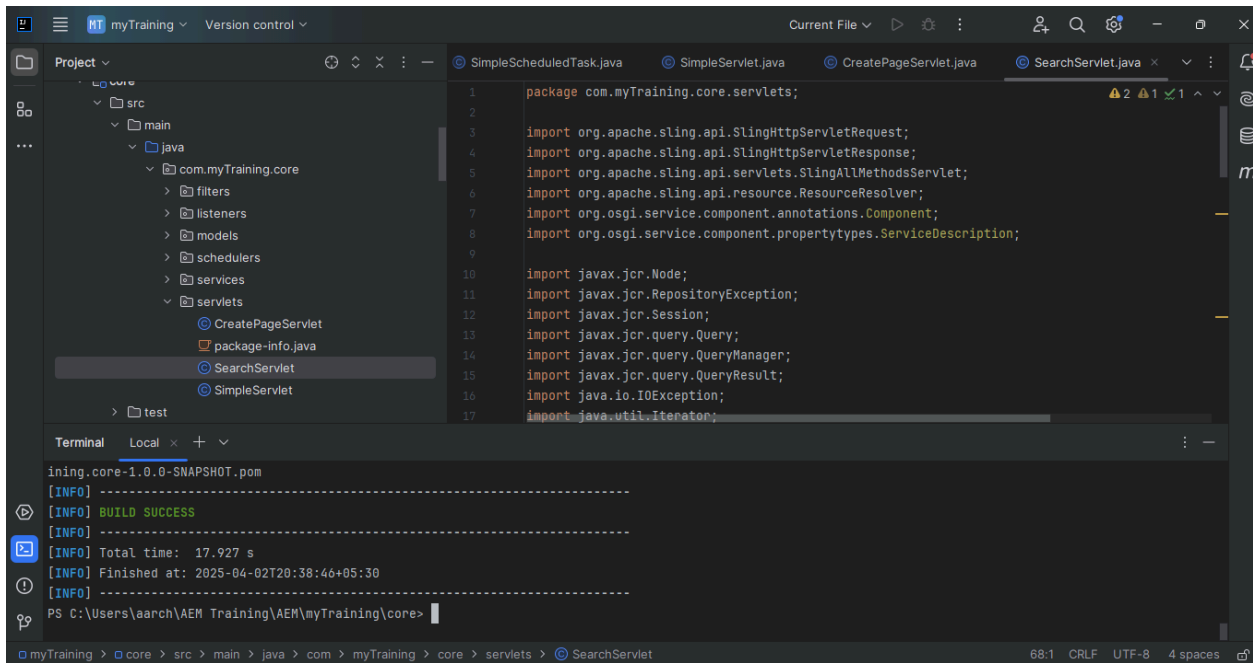
4. Implement **SearchServlet** Using PredicateMap

Objective

- Implement a search functionality in AEM.
- Use **QueryBuilder API** and **PredicateMap** to filter results.

Implementation

1. Extend **SlingSafeMethodsServlet**.
2. Use **@SlingServletPaths** to define the endpoint `/bin/searchcontent`.
3. Construct a **PredicateMap** with search criteria.
4. Fetch search results and return matching page paths.



Search Results:

