

## Create SampleServlet Extending SlingAllMethodsServlet

**Description:** Create an AEM servlet registered by **resourceType** to handle HTTP **GET** requests and log execution.

### Code Snippet:

```
@Component(service = Servlet.class, property = {
    "sling.servlet.resourceTypes=myapp/components/page",
    "sling.servlet.methods=GET"
})

public class SampleServlet extends SlingAllMethodsServlet {

    private static final Logger LOG = LoggerFactory.getLogger(SampleServlet.class);

    @Override
    protected void doGet(SlingHttpServletRequest request, SlingHttpServletResponse response)
        throws ServletException, IOException {
        response.getWriter().write("Sample Servlet Executed");
        LOG.info("Sample Servlet Executed");
    }
}
```

## Create CreatePageServlet Extending SlingSafeMethodsServlet

**Description:** This servlet is registered at a specific path to create pages based on user input. It handles **POST** requests.

### Code Snippet:

```
@Component(service = Servlet.class, property = {
    "sling.servlet.paths=/bin/createPage",
    "sling.servlet.methods=POST"
})

public class CreatePageServlet extends SlingSafeMethodsServlet {

    @Reference
    private ResourceResolverFactory resolverFactory;
```

@Override

```
protected void doPost(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/mysite", pageName,
                "/apps/myapp/templates/page", pageName);
            resolver.commit();
            response.getWriter().write("Page created successfully: " + newPage.getPath());
        } else {
            response.getWriter().write("PageManager is null");
        }
    } catch (PersistenceException e) {
        response.getWriter().write("Page creation failed: " + e.getMessage());
    }
}
```

## Use PageManager API to Create Pages

**Description:** Demonstrates how to dynamically create pages using the **PageManager** API.

### Steps:

1. Retrieve the **PageManager** instance from the ResourceResolver.
2. Call the **create()** method, passing the **parent path**, **page name**, **template path**, and **title**.
3. **Commit** the changes using `resolver.commit()`.

Code Snippet:

```
PageManager pageManager = resolver.adaptTo(PageManager.class);
if (pageManager != null) {
    Page newPage = pageManager.create("/content/mysite", pageName,
        "/apps/myapp/templates/page", pageName);
    resolver.commit();}
```

## Create SearchServlet Using PredicateMap

**Description:** This AEM servlet searches for content using the **QueryBuilder API** and **PredicateMap**.

### Code Snippet:

```
@Component(service = Servlet.class, property = {
    "sling.servlet.paths=/bin/searchContent",
    "sling.servlet.methods=GET"
})

public class SearchServlet extends SlingSafeMethodsServlet {

    @Reference
    private QueryBuilder queryBuilder;

    @Override
    protected void doGet(SlingHttpServletRequest request, SlingHttpServletResponse response)
        throws ServletException, IOException {
        Map<String, String> map = new HashMap<>();
        map.put("path", "/content/mysite");
        map.put("type", "cq:Page");
        map.put("p.limit", "-1");

        Query query = queryBuilder.createQuery(PredicateGroup.create(map),
            request.getResourceResolver().adaptTo(Session.class));

        SearchResult result = query.getResult();
        for (Hit hit : result.getHits()) {
            response.getWriter().write(hit.getPath() + "\n");
        }
    }
}
```