**Sling Servlet**

Sling servlet is nothing but a servlet, which aware of a sling environment, sling resolution, sling request, sling response methods.

A sling servlet can extend either SlingSafeMethodsServlet or SlingAllMethodsServlet.

Difference between SlingSafeMethodsServlet and SlingAllMethodsServlet.

**SlingSafeMethodsServlet –** Extends GenericServlet and only supports GET methods. Can just get the data from the server. Can’t create or post anything on the server.

It has all safe methods like doGet(), doHead(), doOptions()

**SlingAllMethodsServlet –** Extends SlingSafeMethodsServlet and only supports POST methods.

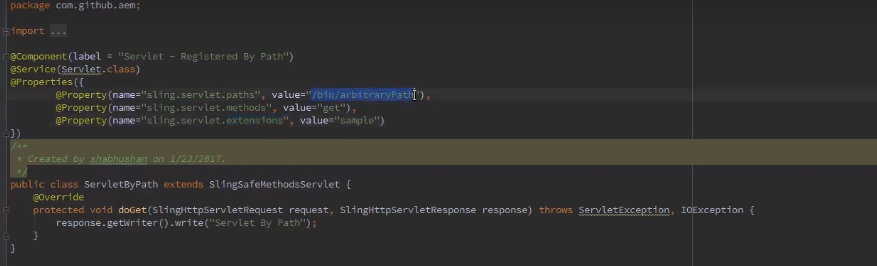
It has methods like doPost().

**Note:** The service method in GenericServlet class will invoke the doGet() or doPost() methods based on the request method.

Servlet can be registered in 3 ways.

1. Using “path” of the servlet
2. Using “resourceType”
3. Using selector

**Method 1: Using path of the servlet**



**@Component –** Defines that the servlet is OSGI component and its life cycle is maintained by OSGI.

**@Service** – Defines that the servlet is OSGI service. But not just any OSGI service, it will be service implementation for Servlet.class interface which will have service method for calling appropriate methods.

If the method is get then the service method will call doGet() method, if its post then it will call doPost() method.

@Properties –Defines properties of the service.

Sling.servlet.path – Servlet will be registed with the value we provide for this property name

Sling.servlet.methods – Methods supported by the servlet

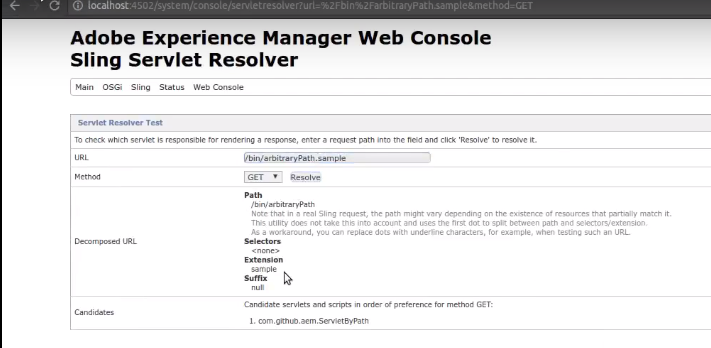
Sling.servlet.extensions – One other option and when we provide this extension value, the servlet path becomes, <http://localhost:4502/bin/servletd/testing.extensionvaluegoeshere>

Console for checking the servlet resolution

Go to the following console, and put the servlet path and verify if it resolves to servlet.

<http://localhost:4502/system/console/serveltresolver>

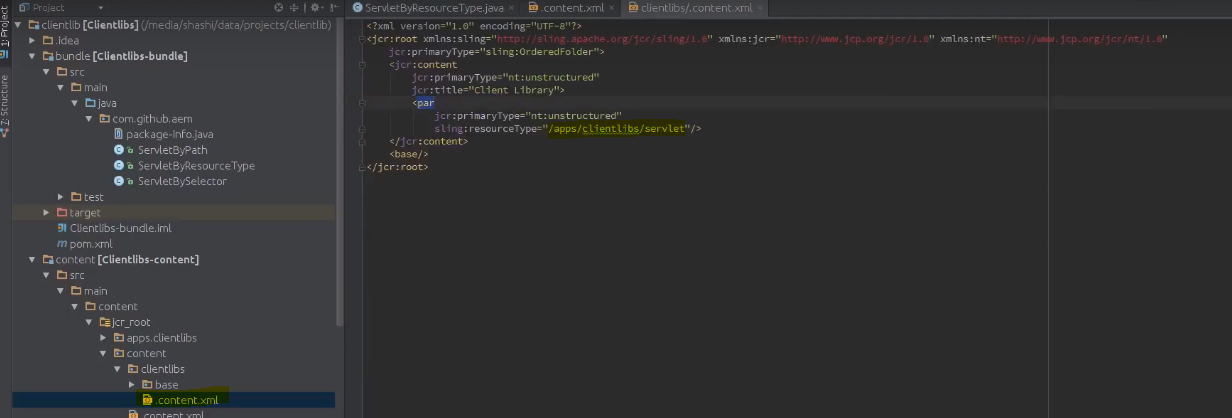


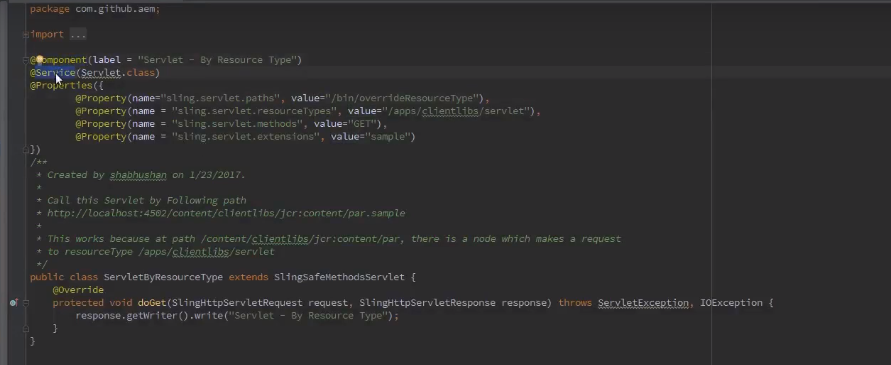


**Method 2: Using resourcetype**

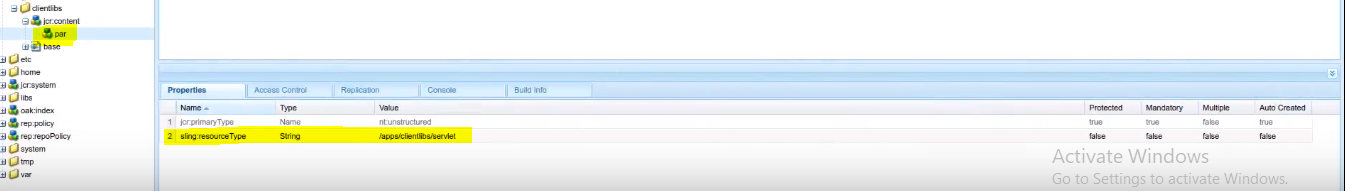
Additionally we will have one more property “sling.servlet.resourceTypes”. This resourceTypes values specifies the list of resourceTypes the servlet want to serve the response.

Ex: If a node resource type is matching with any of the specified resourceTypes in the servlet, then the request would be handled by the servlet and sends the response back.





As the node resource type is matching with the servlet resource type, servlet would be invoked to process the request.

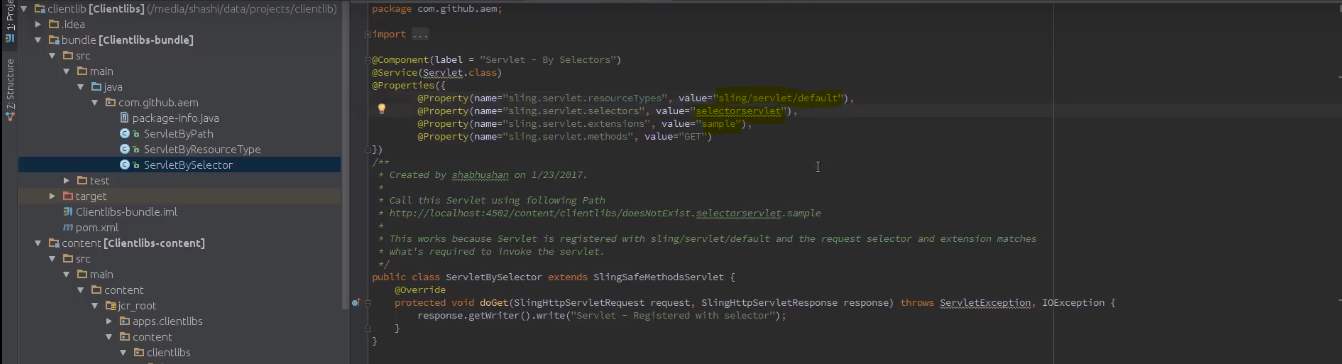


When this node is requested via this URL, <http://localhost:4502/content/clientlibs/jcr:content/par.sample> the servlet would be invoked and response will be returned.

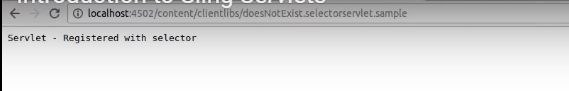
Note: If there is a conflict between paths and resouceTypes, **paths** will always takes the precedence.

**Method 3: Using selector**

The resource types of this kind of servlets should be **“sling/servlet/default”.** This means that this servlet is not bound to any specific resource types, can serve request to all resource types. This made this servlet to invoke it by just the selector.



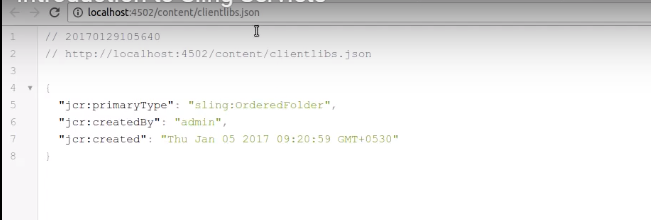
This servlet will be invoked by any request, which has this specified selector (in this case, it is **selectorservlet**) and the extension (in this case, it is **sample**).



AEM uses this kind of servlet to serve many of the requests.

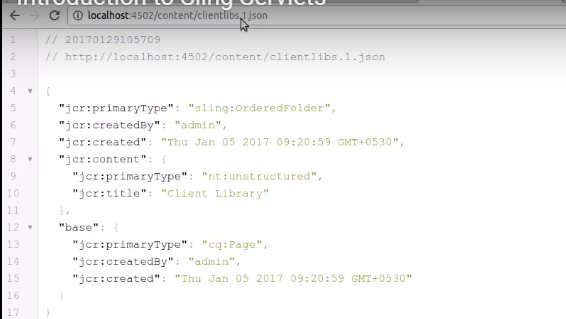
Ex1 : <http://localhost:4502/content/clientlibs.json>

So here, we are requesting the resource in json format (here json is the extension). Which serve the resource response in a JSON format.



Ex2 : <http://localhost:4502/content/clientlibs.1.json> (requesting all the childs one level down to the node in json format)

Here, 1 is the selector and json is the extension.



Ex3 : <http://localhost:4502/content/clientlibs.infinity.json> (requesting the json hierarchy till the end)

Selector : infinity

Extension : json

All the node’s representation in json format would be returned.

