# Sling Servlets

|  |  |  |
| --- | --- | --- |
|  | Type | Description |
| Sling.servlet.paths | (Single)String or String[] | Path under which servlet has to be registered |
| Sling.servlet.resourceTypes | (Single)String or String[] | ResourceTypes for which this servlet will be invoked |
| Sling.servlet.resourceSuperType | (Single) valued String |  |
| Sling.servlet.selectors | (Single)String or String[] | Used in conjuction with Sling.servlet.resourceTypes. |
| Sling.servlet.extension | (Single)String or String[] | Used in conjuction with Sling.servlet.resourceTypes. |
| Sling.servlet.methods | (Single)String or String[] | Used in conjuction with Sling.servlet.resourceTypes. Defaults to GET, HEAD |

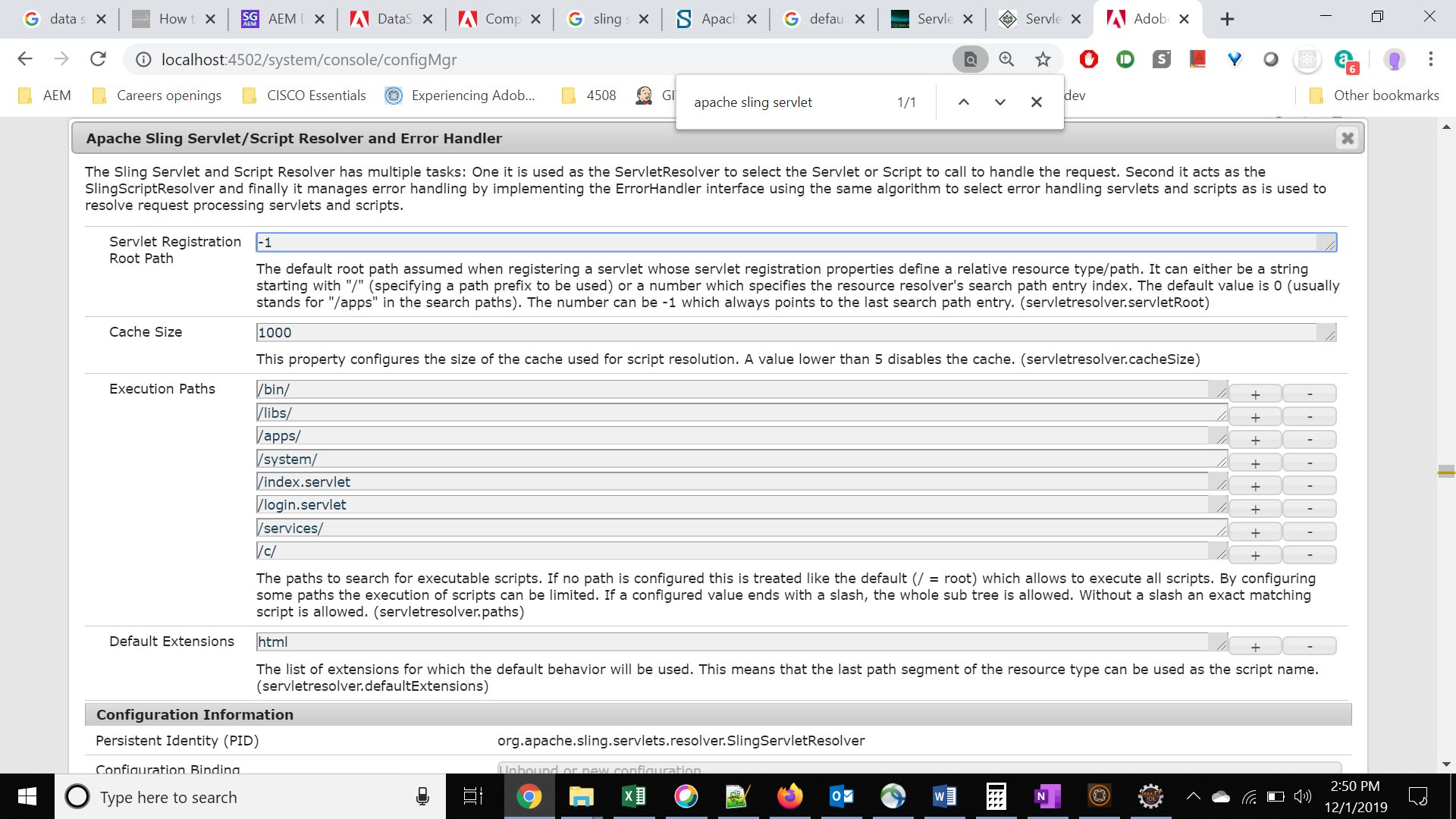
For a Servlet registered as an OSGi service to be used by the Sling Servlet Resolver, either one or both of the sling.servlet.paths or the sling.servlet.resourceTypes

**[Caveats when binding servlets by path](https://sling.apache.org/documentation/the-sling-engine/servlets.html" \l "caveats-when-binding-servlets-by-path)**

Binding servlets by paths has several disadvantages when compared to binding by resource types, namely:

* path-bound servlets cannot be access controlled using the default JCR repository ACLs
* path-bound servlets can only be registered to a path and not a resource type (i.e. no suffix handling)
* if a path-bound servlet is not active, e.g. if the bundle is missing or not started, a POST might result in unexpected results. usually creating a node at /bin/xyz which subsequently overlays the servlets path binding – c node formation in wem
* the mapping is not transparent to a developer looking just at the repository

1. we will have to register the new path of servlet in **Apache Sling Servet/Script Resolver and Error Handler** configuration. Or the servlet will not be functional.



1. we will have to make changes at dispatcher level to allow request to servlet at that path.

Given these drawbacks it is strongly recommended to bind servlets to resource types rather than paths

NOTE:

Using resourceType for registering a servlet is suggested because the sling engine takes care of permissions where as by ‘*resourcePaths*’ wont (as suggested in first point under [**Caveats when binding servlets by path**](https://sling.apache.org/documentation/the-sling-engine/servlets.html#caveats-when-binding-servlets-by-path))

* *Another good reason to use resourceType is that the Sling Engine will take care of permissions for you. Users who cannot access a particular resource will not be able to invoke the servlet.*

# Difference between Sling Servlet & Java Servlet

Java Servlet :

The servlet container after initializing the servlet calls the init() method exactly once .

In Sling environment:

1. The service provided initialized the servlet instance
2. Sling servlet Resolver picks up the instances , initializes and destroys them as needed.

# [Servlet Resolution Order](https://sling.apache.org/documentation/the-sling-engine/servlets.html" \l "servlet-resolution-order)

The following order rules are being followed when trying to resolve a servlet for a given request URL and request method and multiple candidates would match. Then the following candidate is being picked (if one rule doesn't lead to one winner, the next rule is being evaluated):

1. The one with the highest number of matching selectors + extension
2. The one which is registered to a resource type closest to the requested one (when traversing the resource type hierarchy up)
3. The one with the highest service.ranking property