Documentation
Main
Getting Started
The Sling Engine
Development
Bundles
Tutorials & How-Tos
Maven Plugins
Configuration

API Docs
Sling 11
Sling 10
Sling 9
All versions

Support
Wiki
FAQ
Site Map

Project Info
Downloads
License
News
Releases
Issue Tracker
Links
Contributing
Project Information
Security

Source
GitHub
Git at Apache

Apache Software Foundation
Thanks!
Become a Sponsor
Buy Stuff

Dispatching Requests

Main process

The following steps should give you an overview how a request is processed in Sling. Details can be found under provided links.

1. The client sends the request
2. This step applies only if a Servlet Container is installed and Sling is embedded: Servlet Container gets request and forwards to OSGi HttpService
3. OSGi HttpService looks for responsible registered Servlet or resource (see 102.4 of the OSGi compendium)
4. OSGi HttpService calls handleSecurity of the HttpContext associated with the servlet/resource. In case of Sling this calls into SlingMainServlet.handleSecurity and then into SlingAuthenticator.authenticate
    1. SlingAuthenticator selects an authentication handler for the request and forwards the authenticate call. On success a javax.jcr.Session is created, the request attributes required by the HTTP Service spec are set (like org.osgi.service.http.authentication.remote.user and org.osgi.service.http.authentication.typeand also the javax.jcr.Session which is used later is set in the request attributes. On success, continue with step 5.
    2. If authentication fails either an anonymous session is acquired (if anonymous is allowed per configuration) or the login method is called. If anonymous is allowed, continue with step 5.
    3. The login method selects an AuthenticationHandler and forwards the login call to the AuthenticationHandler.requestAuthentication method to cause the client to authenticate. Request processing stops here (SlingMainServlet.handleSecurity returns false).
5. After getting a response the HttpService either terminates the request (if authentication failed and SlingMainServlet.handleSecurity returned false) or continues by either spooling the resource or in the case of Sling calling the SlingMainServlet.service method.
6. The SlingMainServlet.service method is the entry point into the Sling proper. This method sets up the request:
    - Wraps the HttpServletRequest and the HttpServletResponse into the SlingHttpServletRequest and the SlingHttpServletResponse
    - Checks if Sling is ready for processing the request (checks at the moment for an existing ResourceResolverFactory service, a ServletResolver service and a MimeTypeService)
    - Create the ResourceResolver based on the Session (by default creates a JcrResourceResolver2)
    - Locate the Resource on the basis of the request by calling ResourceResovler.resolve through RequestData.initResource (see also URL decomposition)
    - Locate the servlet or script (see Servlets) by calling ServletResolver.resolveServlet through RequestData.initServlet
7. After this setup, the request level filters are called (the ones registered as javax.servlet.Filter with the property filter.scope=request, see Filters for details). If any called filter doesn't call FilterChain.doFilter at the end of the Filter.doFilter method request processing stops here.
8. After having called all request level filters, the component level filters (registered with the property filter.scope=component, see Filters for details) are called.
9. After having called the component level filters, the request servlet or script is finally called to process the request.

Include/Forward

If a servlet or script is including another resource for processing through the RequestDispatcher.include or RequestDispatcher.forward (or any JSP or feature of another scripting language which relies on one of this two methods) the following processing takes place:

1. Code in the processing servlet or script calls RequestDispatcher.include or RequestDispatcher.forward.
2. The resource is resolved though ResourceResolver.getResource (if the RequestDispatcher has not been created with a resource already)
3. The servlet or script to handle the resource is resolved calling the ServletResolver.resolverServlet method.
4. The component level filters (registered with the property filter.scope=component) are called again (see Filters for details).
5. The servlet or script is called to process the request.

Note that these steps are processed for every include or forward call.

Included Request Attributes

When servlet or script is called as a result of RequestDispatcher.include the following request attributes are set:

Attribute Name **Attribute Type**	**Description**
org.apache.sling.api.include.servlet javax.servlet.Servlet	The name of the request attribute containing the Servlet which included the servlet currently being active.
org.apache.sling.api.include.resource org.apache.sling.api.resource.Resource	The name of the request attribute containing the Resource underlying the Servlet which included the servlet currently being active.
org.apache.sling.api.include.request_path_info org.apache.sling.api.request.RequestPathInfo	The name of the request attribute containing the RequestPathInfo underlying the Servlet which included the servlet currently being active
javax.servlet.include.request_uri String	The name of the request attribute containing the HttpServletRequest.getRequestURI() of the request which included the servlet currently being active underlying the Servlet which included the servlet currently being active. **Note:** In Sling, the HttpServletRequest.getRequestURI() method will always return the same result regardless of whether it is called from the client request processing servlet or script or from an included servlet or script. This request attribute is set for compatibility with the Servlet API specification.
javax.servlet.include.context_path String	The name of the request attribute containing the HttpServletRequest.getContextPath() of the request which included the servlet currently being active underlying the Servlet which included the servlet currently being active. **Note:** In Sling, the HttpServletRequest.getContextPath() method will always return the same result regardless of whether it is called from the client request processing servlet or script or from an included servlet or script. This request attribute is set for compatibility with the Servlet API specification.
javax.servlet.include.servlet_path String	The name of the request attribute containing the HttpServletRequest.getServletPath() of the request which included the servlet currently being active underlying the Servlet which included the servlet currently being active. **Note:** In Sling, the HttpServletRequest.getServletPath() method will always return the same result regardless of whether it is called from the client request processing servlet or script or from an included servlet or script. This request attribute is set for compatibility with the Servlet API specification.
javax.servlet.include.path_info String	The name of the request attribute containing the HttpServletRequest.getPathInfo() of the request which included the servlet currently being active underlying the Servlet which included the servlet currently being active. **Note:** In Sling, the HttpServletRequest.getPathInfo() method will always return the same result regardless of whether it is called from the client request processing servlet or script or from an included servlet or script. This request attribute is set for compatibility with the Servlet API specification.
javax.servlet.include.query_string String	The name of the request attribute containing the HttpServletRequest.getQueryString() of the request which included the servlet currently being active underlying the Servlet which included the servlet currently being active. **Note:** In Sling, the HttpServletRequest.getQueryString() method will always return the same result regardless of whether

Attribute Name Attribute Type	Description
	it is called from the client request processing servlet or script or from an included servlet or script. This request attribute is set for compatibility with the Servlet API specification.

Constants are defined in the org.apache.sling.api.SlingConstants class for these request attributes.

Note: These request attributes are not set if the servlet or script is called to handle the request or as a result of RequestDispatcher.forward.

Last modified by Konrad Windszus on Fri Jul 13 11:08:10 2018 +0200