<?xml version="1.0" encoding="UTF-8"?>

<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee

http://xmlns.jcp.org/xml/ns/javaee/web-app\_3\_1.xsd"

metadata-complete="true"

version="3.1">

<display-name>Confluence</display-name>

<description>Confluence Web App</description>

<absolute-ordering />

<context-param>

<param-name>contextConfigLocation</param-name>

<param-value>com.atlassian.confluence.setup.ConfluenceAppConfig</param-value>

</context-param>

<context-param>

<param-name>contextClass</param-name>

<param-value>com.atlassian.confluence.setup.ConfluenceAnnotationConfigWebApplicationContext</param-value>

</context-param>

<!-- Uncomment the following to disable the space export long running task. -->

<!--

<context-param>

<param-name>unsupportedContainersForExportLongRunningTask</param-name>

<param-value>websphere,jboss</param-value>

</context-param>

-->

<filter>

<filter-name>debug-before-request</filter-name>

<filter-class>com.atlassian.confluence.web.filter.DebugFilter</filter-class>

<init-param>

<param-name>phase</param-name>

<param-value>before</param-value>

</init-param>

<init-param>

<param-name>dispatcher</param-name>

<param-value>REQUEST</param-value>

</init-param>

</filter>

<filter>

<filter-name>debug-before-include</filter-name>

<filter-class>com.atlassian.confluence.web.filter.DebugFilter</filter-class>

<init-param>

<param-name>phase</param-name>

<param-value>before</param-value>

</init-param>

<init-param>

<param-name>dispatcher</param-name>

<param-value>INCLUDE</param-value>

</init-param>

</filter>

<filter>

<filter-name>debug-before-forward</filter-name>

<filter-class>com.atlassian.confluence.web.filter.DebugFilter</filter-class>

<init-param>

<param-name>phase</param-name>

<param-value>before</param-value>

</init-param>

<init-param>

<param-name>dispatcher</param-name>

<param-value>FORWARD</param-value>

</init-param>

</filter>

<filter>

<filter-name>debug-before-error</filter-name>

<filter-class>com.atlassian.confluence.web.filter.DebugFilter</filter-class>

<init-param>

<param-name>phase</param-name>

<param-value>before</param-value>

</init-param>

<init-param>

<param-name>dispatcher</param-name>

<param-value>ERROR</param-value>

</init-param>

</filter>

<filter>

<filter-name>debug-after-request</filter-name>

<filter-class>com.atlassian.confluence.web.filter.DebugFilter</filter-class>

<init-param>

<param-name>phase</param-name>

<param-value>after</param-value>

</init-param>

<init-param>

<param-name>dispatcher</param-name>

<param-value>REQUEST</param-value>

</init-param>

</filter>

<filter>

<filter-name>debug-after-include</filter-name>

<filter-class>com.atlassian.confluence.web.filter.DebugFilter</filter-class>

<init-param>

<param-name>phase</param-name>

<param-value>after</param-value>

</init-param>

<init-param>

<param-name>dispatcher</param-name>

<param-value>INCLUDE</param-value>

</init-param>

</filter>

<filter>

<filter-name>debug-after-forward</filter-name>

<filter-class>com.atlassian.confluence.web.filter.DebugFilter</filter-class>

<init-param>

<param-name>phase</param-name>

<param-value>after</param-value>

</init-param>

<init-param>

<param-name>dispatcher</param-name>

<param-value>FORWARD</param-value>

</init-param>

</filter>

<filter>

<filter-name>debug-after-error</filter-name>

<filter-class>com.atlassian.confluence.web.filter.DebugFilter</filter-class>

<init-param>

<param-name>phase</param-name>

<param-value>after</param-value>

</init-param>

<init-param>

<param-name>dispatcher</param-name>

<param-value>ERROR</param-value>

</init-param>

</filter>

<filter>

<filter-name>webwork-cleanup</filter-name>

<filter-class>com.atlassian.confluence.plugin.servlet.filter.ActionContextCleanUp</filter-class>

</filter>

<filter>

<filter-name>header-sanitiser</filter-name>

<filter-class>com.atlassian.core.filters.HeaderSanitisingFilter</filter-class>

</filter>

<filter>

<filter-name>log404s</filter-name>

<filter-class>com.atlassian.confluence.servlet.FourOhFourErrorLoggingFilter</filter-class>

</filter>

<filter>

<filter-name>jmx</filter-name>

<filter-class>com.atlassian.confluence.jmx.JmxFilter</filter-class>

</filter>

<filter>

<filter-name>request-param-cleaner</filter-name>

<filter-class>com.atlassian.confluence.web.filter.validateparam.RequestParamValidationFilter</filter-class>

<init-param>

<param-name>whitelistStrategy</param-name>

<param-value>defaultRequestParamCleanerWhitelistStrategy</param-value>

</init-param>

</filter>

<!-- Work around classloader issues between core and plugins by disabling async request support in spring -->

<!-- TODO replace with an implementation that enforces only plugins classloaders registering spring filters and servlets -->

<filter>

<filter-name>ignore-webasyncmanager</filter-name>

<filter-class>com.atlassian.confluence.internal.web.filter.spring.IgnoreWebAsyncManagerFilter</filter-class>

</filter>

<!--USE WITH CAUTION!

This filter will drop any request that comes in when Confluence is not finished setting up.

To prevent this from dropping valid requests ensure that the filter url mapping maps to exactly the request you want

to drop.-->

<filter>

<filter-name>dropIfNotSetupFilter</filter-name>

<filter-class>com.atlassian.confluence.web.filter.DropIfNotSetupFilter</filter-class>

</filter>

<filter>

<filter-name>httpRequestMonitoringFilter</filter-name>

<filter-class>com.atlassian.confluence.internal.diagnostics.HttpRequestMonitoringFilter</filter-class>

</filter>

<filter>

<filter-name>language</filter-name>

<filter-class>com.atlassian.confluence.web.filter.LanguageExtractionFilter</filter-class>

</filter>

<filter>

<filter-name>translation-mode</filter-name>

<filter-class>com.atlassian.confluence.web.filter.TranslationModeFilter</filter-class>

</filter>

<filter>

<filter-name>johnson</filter-name>

<filter-class>com.atlassian.confluence.web.ConfluenceJohnsonFilter</filter-class>

</filter>

<filter>

<filter-name>sessioninview</filter-name>

<filter-class>com.atlassian.confluence.web.filter.ConfluenceOpenSessionInViewFilter</filter-class>

</filter>

<filter>

<filter-name>login</filter-name>

<filter-class>com.atlassian.seraph.filter.LoginFilter</filter-class>

<init-param>

<!-- This is required to opt-in for os\_username and os\_password in Seraph >= 3.1.0 -->

<param-name>allowUrlParameterValue</param-name>

<param-value>true</param-value>

</init-param>

</filter>

<filter>

<filter-name>authenticator-metrics</filter-name>

<filter-class>com.atlassian.confluence.impl.seraph.AuthenticatorMetricsFilter</filter-class>

</filter>

<filter>

<filter-name>trustedapp</filter-name>

<filter-class>com.atlassian.confluence.security.trust.seraph.ConfluenceTrustedApplicationsFilter</filter-class>

</filter>

<filter>

<filter-name>zipkinFilter</filter-name>

<filter-class>com.atlassian.confluence.web.filter.ZipkinTracingFilter</filter-class>

</filter>

<filter>

<filter-name>requestcache</filter-name>

<filter-class>com.atlassian.confluence.util.RequestCacheThreadLocalFilter</filter-class>

</filter>

<filter>

<filter-name>messagesDecoratorFilter</filter-name>

<filter-class>com.atlassian.confluence.util.message.MessagesDecoratorFilter</filter-class>

</filter>

<filter>

<filter-name>sitemesh</filter-name>

<filter-class>com.atlassian.confluence.util.profiling.ProfilingSiteMeshFilter</filter-class>

</filter>

<filter>

<filter-name>sitemesh-error</filter-name>

<filter-class>com.atlassian.confluence.util.profiling.ProfilingSiteMeshFilter</filter-class>

<init-param>

<param-name>dispatcher</param-name>

<param-value>ERROR</param-value>

</init-param>

</filter>

<filter>

<filter-name>encoding</filter-name>

<filter-class>com.atlassian.confluence.setup.ConfluenceEncodingFilter</filter-class>

</filter>

<filter>

<filter-name>caching</filter-name>

<filter-class>com.atlassian.confluence.web.filter.ConfluenceCachingFilter</filter-class>

</filter>

<filter>

<filter-name>security</filter-name>

<filter-class>com.atlassian.confluence.web.filter.ConfluenceSecurityFilter</filter-class>

</filter>

<filter>

<filter-name>timeout</filter-name>

<filter-class>com.atlassian.confluence.web.filter.ConfluenceTimeoutFilter</filter-class>

<init-param>

<param-name>urlPatternsToExclude</param-name>

<param-value>

/rest/quickreload/\*\*,

/rest/mywork/latest/status/notification/count

</param-value>

</init-param>

</filter>

<filter>

<filter-name>userthreadlocal</filter-name>

<filter-class>com.atlassian.confluence.util.UserThreadLocalFilter</filter-class>

</filter>

<filter>

<filter-name>maueventfilter</filter-name>

<filter-class>com.atlassian.confluence.web.filter.MauEventFilter</filter-class>

</filter>

<filter>

<filter-name>usernameheader</filter-name>

<filter-class>com.atlassian.confluence.util.UserNameHeaderFilter</filter-class>

</filter>

<filter>

<filter-name>servletcontextthreadlocal</filter-name>

<filter-class>com.atlassian.core.filters.ServletContextThreadLocalFilter</filter-class>

</filter>

<filter>

<filter-name>transactionalCacheFactoryCleanupFilter</filter-name>

<filter-class>com.atlassian.confluence.cache.TransactionalCacheFactoryCleanupFilter</filter-class>

</filter>

<filter>

<filter-name>threadLocalCache</filter-name>

<filter-class>com.atlassian.confluence.web.filter.ThreadLocalCacheFilter</filter-class>

</filter>

<filter>

<filter-name>expires-one-hour</filter-name>

<filter-class>com.atlassian.core.filters.ExpiresFilter</filter-class>

<init-param>

<!-- specify the which HTTP parameter to use to turn the filter on or off -->

<!-- if not specified - defaults to "profile.filter" -->

<param-name>expiryTimeInSeconds</param-name>

<param-value>3600</param-value>

</init-param>

</filter>

<filter>

<filter-name>thread-local-error-collection</filter-name>

<filter-class>com.atlassian.confluence.util.ConfluenceErrorFilter</filter-class>

</filter>

<filter>

<filter-name>profiling</filter-name>

<filter-class>com.atlassian.confluence.util.profiling.ConfluenceProfilingFilter</filter-class>

<init-param>

<!-- specify the which HTTP parameter to use to turn the filter on or off -->

<!-- if not specified - defaults to "profile.filter" -->

<param-name>activate.param</param-name>

<param-value>profile</param-value>

</init-param>

<init-param>

<!-- specify the whether to start the filter automatically -->

<!-- if not specified - defaults to "true" -->

<param-name>autostart</param-name>

<param-value>false</param-value>

</init-param>

</filter>

<filter>

<filter-name>AccessLogFilter</filter-name>

<filter-class>com.atlassian.confluence.util.AccessLogFilter</filter-class>

</filter>

<filter>

<filter-name>ClusterHeaderFilter</filter-name>

<filter-class>com.atlassian.confluence.util.ClusterHeaderFilter</filter-class>

</filter>

<!-- this filter is used to rewrite through the /s/\* filter to add caching headers. see: urlrewrite.xml -->

<filter>

<filter-name>UrlRewriteFilter</filter-name>

<filter-class>org.tuckey.web.filters.urlrewrite.UrlRewriteFilter</filter-class>

</filter>

<filter>

<filter-name>LoggingContextFilter</filter-name>

<filter-class>com.atlassian.confluence.util.LoggingContextFilter</filter-class>

</filter>

<filter>

<filter-name>UserLoggingContextFilter</filter-name>

<filter-class>com.atlassian.confluence.util.UserLoggingContextFilter</filter-class>

</filter>

<filter>

<filter-name>RequestTimeFilter</filter-name>

<filter-class>com.atlassian.confluence.core.datetime.RequestTimeThreadLocalFilter</filter-class>

</filter>

<filter>

<filter-name>ResponseOutputStreamFilter</filter-name>

<filter-class>com.atlassian.confluence.web.filter.ResponseOutputStreamFilter</filter-class>

</filter>

<filter>

<filter-name>OpenTenantGateFilter</filter-name>

<filter-class>com.atlassian.confluence.tenant.TenantGateFilter</filter-class>

<init-param>

<param-name>permit</param-name>

<param-value>true</param-value>

</init-param>

</filter>

<filter>

<filter-name>CloseTenantGateFilter</filter-name>

<filter-class>com.atlassian.confluence.tenant.TenantGateFilter</filter-class>

</filter>

<filter>

<filter-name>ServerInfoFilter</filter-name>

<filter-class>com.atlassian.confluence.util.ServerInfoFilter</filter-class>

</filter>

<filter>

<filter-name>MobileAppRequestFilter</filter-name>

<filter-class>com.atlassian.confluence.util.MobileAppRequestFilter</filter-class>

</filter>

<!-- Plugins 2.5 filter changes -->

<filter>

<filter-name>filter-plugin-dispatcher-after-encoding-request</filter-name>

<filter-class>com.atlassian.confluence.plugin.servlet.filter.ServletFilterModuleContainerFilter</filter-class>

<init-param>

<param-name>location</param-name>

<param-value>after-encoding</param-value>

</init-param>

<init-param>

<param-name>dispatcher</param-name>

<param-value>REQUEST</param-value>

</init-param>

</filter>

<filter>

<filter-name>filter-plugin-dispatcher-after-encoding-forward</filter-name>

<filter-class>com.atlassian.confluence.plugin.servlet.filter.ServletFilterModuleContainerFilter</filter-class>

<init-param>

<param-name>location</param-name>

<param-value>after-encoding</param-value>

</init-param>

<init-param>

<param-name>dispatcher</param-name>

<param-value>FORWARD</param-value>

</init-param>

</filter>

<filter>

<filter-name>filter-plugin-dispatcher-after-encoding-include</filter-name>

<filter-class>com.atlassian.confluence.plugin.servlet.filter.ServletFilterModuleContainerFilter</filter-class>

<init-param>

<param-name>location</param-name>

<param-value>after-encoding</param-value>

</init-param>

<init-param>

<param-name>dispatcher</param-name>

<param-value>INCLUDE</param-value>

</init-param>

</filter>

<filter>

<filter-name>filter-plugin-dispatcher-after-encoding-error</filter-name>

<filter-class>com.atlassian.confluence.plugin.servlet.filter.ServletFilterModuleContainerFilter</filter-class>

<init-param>

<param-name>location</param-name>

<param-value>after-encoding</param-value>

</init-param>

<init-param>

<param-name>dispatcher</param-name>

<param-value>ERROR</param-value>

</init-param>

</filter>

<filter>

<filter-name>filter-plugin-dispatcher-before-login-request</filter-name>

<filter-class>com.atlassian.confluence.plugin.servlet.filter.ServletFilterModuleContainerFilter</filter-class>

<init-param>

<param-name>location</param-name>

<param-value>before-login</param-value>

</init-param>

<init-param>

<param-name>dispatcher</param-name>

<param-value>REQUEST</param-value>

</init-param>

</filter>

<filter>

<filter-name>filter-plugin-dispatcher-before-login-forward</filter-name>

<filter-class>com.atlassian.confluence.plugin.servlet.filter.ServletFilterModuleContainerFilter</filter-class>

<init-param>

<param-name>location</param-name>

<param-value>before-login</param-value>

</init-param>

<init-param>

<param-name>dispatcher</param-name>

<param-value>FORWARD</param-value>

</init-param>

</filter>

<filter>

<filter-name>filter-plugin-dispatcher-before-login-include</filter-name>

<filter-class>com.atlassian.confluence.plugin.servlet.filter.ServletFilterModuleContainerFilter</filter-class>

<init-param>

<param-name>location</param-name>

<param-value>before-login</param-value>

</init-param>

<init-param>

<param-name>dispatcher</param-name>

<param-value>INCLUDE</param-value>

</init-param>

</filter>

<filter>

<filter-name>filter-plugin-dispatcher-before-login-error</filter-name>

<filter-class>com.atlassian.confluence.plugin.servlet.filter.ServletFilterModuleContainerFilter</filter-class>

<init-param>

<param-name>location</param-name>

<param-value>before-login</param-value>

</init-param>

<init-param>

<param-name>dispatcher</param-name>

<param-value>ERROR</param-value>

</init-param>

</filter>

<filter>

<filter-name>filter-plugin-dispatcher-before-decoration-request</filter-name>

<filter-class>com.atlassian.confluence.plugin.servlet.filter.ServletFilterModuleContainerFilter</filter-class>

<init-param>

<param-name>location</param-name>

<param-value>before-decoration</param-value>

</init-param>

<init-param>

<param-name>dispatcher</param-name>

<param-value>REQUEST</param-value>

</init-param>

</filter>

<filter>

<filter-name>filter-plugin-dispatcher-before-decoration-forward</filter-name>

<filter-class>com.atlassian.confluence.plugin.servlet.filter.ServletFilterModuleContainerFilter</filter-class>

<init-param>

<param-name>location</param-name>

<param-value>before-decoration</param-value>

</init-param>

<init-param>

<param-name>dispatcher</param-name>

<param-value>FORWARD</param-value>

</init-param>

</filter>

<filter>

<filter-name>filter-plugin-dispatcher-before-decoration-include</filter-name>

<filter-class>com.atlassian.confluence.plugin.servlet.filter.ServletFilterModuleContainerFilter</filter-class>

<init-param>

<param-name>location</param-name>

<param-value>before-decoration</param-value>

</init-param>

<init-param>

<param-name>dispatcher</param-name>

<param-value>INCLUDE</param-value>

</init-param>

</filter>

<filter>

<filter-name>filter-plugin-dispatcher-before-decoration-error</filter-name>

<filter-class>com.atlassian.confluence.plugin.servlet.filter.ServletFilterModuleContainerFilter</filter-class>

<init-param>

<param-name>location</param-name>

<param-value>before-decoration</param-value>

</init-param>

<init-param>

<param-name>dispatcher</param-name>

<param-value>ERROR</param-value>

</init-param>

</filter>

<filter>

<filter-name>filter-plugin-dispatcher-before-dispatch-request</filter-name>

<filter-class>com.atlassian.confluence.plugin.servlet.filter.ServletFilterModuleContainerFilter</filter-class>

<init-param>

<param-name>location</param-name>

<param-value>before-dispatch</param-value>

</init-param>

<init-param>

<param-name>dispatcher</param-name>

<param-value>REQUEST</param-value>

</init-param>

</filter>

<filter>

<filter-name>filter-plugin-dispatcher-before-dispatch-forward</filter-name>

<filter-class>com.atlassian.confluence.plugin.servlet.filter.ServletFilterModuleContainerFilter</filter-class>

<init-param>

<param-name>location</param-name>

<param-value>before-dispatch</param-value>

</init-param>

<init-param>

<param-name>dispatcher</param-name>

<param-value>FORWARD</param-value>

</init-param>

</filter>

<filter>

<filter-name>filter-plugin-dispatcher-before-dispatch-include</filter-name>

<filter-class>com.atlassian.confluence.plugin.servlet.filter.ServletFilterModuleContainerFilter</filter-class>

<init-param>

<param-name>location</param-name>

<param-value>before-dispatch</param-value>

</init-param>

<init-param>

<param-name>dispatcher</param-name>

<param-value>INCLUDE</param-value>

</init-param>

</filter>

<filter>

<filter-name>filter-plugin-dispatcher-before-dispatch-error</filter-name>

<filter-class>com.atlassian.confluence.plugin.servlet.filter.ServletFilterModuleContainerFilter</filter-class>

<init-param>

<param-name>location</param-name>

<param-value>before-dispatch</param-value>

</init-param>

<init-param>

<param-name>dispatcher</param-name>

<param-value>ERROR</param-value>

</init-param>

</filter>

<filter>

<description>Manages the VCache request context</description>

<filter-name>vcache-request-context</filter-name>

<filter-class>com.atlassian.confluence.impl.vcache.VCacheRequestContextFilter</filter-class>

</filter>

<filter>

<filter-name>legacyRemoteApiEventPublishingFilter</filter-name>

<filter-class>org.springframework.web.filter.DelegatingFilterProxy</filter-class>

</filter>

<filter-mapping>

<filter-name>debug-before-request</filter-name>

<url-pattern>/\*</url-pattern>

<dispatcher>REQUEST</dispatcher>

</filter-mapping>

<filter-mapping>

<filter-name>debug-before-include</filter-name>

<url-pattern>/\*</url-pattern>

<dispatcher>INCLUDE</dispatcher>

</filter-mapping>

<filter-mapping>

<filter-name>debug-before-forward</filter-name>

<url-pattern>/\*</url-pattern>

<dispatcher>FORWARD</dispatcher>

</filter-mapping>

<filter-mapping>

<filter-name>debug-before-error</filter-name>

<url-pattern>/\*</url-pattern>

<dispatcher>ERROR</dispatcher>

</filter-mapping>

<!-- End plugins 2.5 filter changes -->

<!--

These mappings for dropIfNotSetupFilter are hacks to prevent setup from failing if there are Confluence browser

tabs open that are pointing to the same instance. They should be safe to remove once CONFDEV-9627 is fixed.

-->

<filter-mapping>

<filter-name>dropIfNotSetupFilter</filter-name>

<url-pattern>/json/startheartbeatactivity.action</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>dropIfNotSetupFilter</filter-name>

<url-pattern>/rest/tinymce/\*</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>dropIfNotSetupFilter</filter-name>

<url-pattern>/rest/quickreload/\*</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>dropIfNotSetupFilter</filter-name>

<url-pattern>/rest/analytics/\*</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>dropIfNotSetupFilter</filter-name>

<url-pattern>/rest/synchrony-interop/\*</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>dropIfNotSetupFilter</filter-name>

<url-pattern>/rest/mywork/\*</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>httpRequestMonitoringFilter</filter-name>

<url-pattern>/\*</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>log404s</filter-name>

<url-pattern>/\*</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>header-sanitiser</filter-name>

<url-pattern>/\*</url-pattern>

</filter-mapping>

<!-- THIS MUST REMAIN AS THE TOP FILTER SO THAT THE ENCODING CAN BE SET BEFORE ANYTHING ELSE TOUCHES IT (well, except the header sanitiser)-->

<filter-mapping>

<filter-name>encoding</filter-name>

<url-pattern>\*.vm</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>encoding</filter-name>

<url-pattern>/display/\*</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>encoding</filter-name>

<url-pattern>\*.action</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>encoding</filter-name>

<url-pattern>/download/\*</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>encoding</filter-name>

<url-pattern>/plugins/servlet/\*</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>encoding</filter-name>

<url-pattern>/label/\*</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>encoding</filter-name>

<url-pattern>/s/\*</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>johnson</filter-name>

<url-pattern>\*.jsp</url-pattern>

<dispatcher>REQUEST</dispatcher>

<dispatcher>FORWARD</dispatcher>

</filter-mapping>

<filter-mapping>

<filter-name>johnson</filter-name>

<url-pattern>\*.vm</url-pattern>

<dispatcher>REQUEST</dispatcher>

<dispatcher>FORWARD</dispatcher>

</filter-mapping>

<filter-mapping>

<filter-name>johnson</filter-name>

<url-pattern>/display/\*</url-pattern>

<dispatcher>REQUEST</dispatcher>

<dispatcher>FORWARD</dispatcher>

</filter-mapping>

<filter-mapping>

<filter-name>johnson</filter-name>

<url-pattern>/download/\*</url-pattern>

<dispatcher>REQUEST</dispatcher>

<dispatcher>FORWARD</dispatcher>

</filter-mapping>

<filter-mapping>

<filter-name>johnson</filter-name>

<url-pattern>/label/\*</url-pattern>

<dispatcher>REQUEST</dispatcher>

<dispatcher>FORWARD</dispatcher>

</filter-mapping>

<filter-mapping>

<filter-name>johnson</filter-name>

<url-pattern>\*.action</url-pattern>

<dispatcher>REQUEST</dispatcher>

<dispatcher>FORWARD</dispatcher>

</filter-mapping>

<filter-mapping>

<filter-name>johnson</filter-name>

<url-pattern>/plugins/servlet/\*</url-pattern>

<dispatcher>REQUEST</dispatcher>

<dispatcher>FORWARD</dispatcher>

</filter-mapping>

<filter-mapping>

<filter-name>ResponseOutputStreamFilter</filter-name>

<url-pattern>/\*</url-pattern>

</filter-mapping>

<!-- Must come before requestcache -->

<filter-mapping>

<filter-name>zipkinFilter</filter-name>

<url-pattern>/\*</url-pattern>

<dispatcher>REQUEST</dispatcher>

<dispatcher>ERROR</dispatcher>

</filter-mapping>

<filter-mapping>

<filter-name>requestcache</filter-name>

<url-pattern>/\*</url-pattern>

<dispatcher>REQUEST</dispatcher>

<dispatcher>ERROR</dispatcher>

</filter-mapping>

<filter-mapping>

<filter-name>LoggingContextFilter</filter-name>

<url-pattern>/\*</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>vcache-request-context</filter-name>

<url-pattern>/\*</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>language</filter-name>

<url-pattern>/\*</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>webwork-cleanup</filter-name>

<url-pattern>/\*</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>translation-mode</filter-name>

<url-pattern>/\*</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>request-param-cleaner</filter-name>

<url-pattern>/\*</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>ignore-webasyncmanager</filter-name>

<url-pattern>/\*</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>ServerInfoFilter</filter-name>

<url-pattern>/server-info.action</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>MobileAppRequestFilter</filter-name>

<url-pattern>/\*</url-pattern>

</filter-mapping>

<!-- code that has to do tenant setup -->

<filter-mapping>

<filter-name>OpenTenantGateFilter</filter-name>

<url-pattern>/admin/appTrustCertificate</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>OpenTenantGateFilter</filter-name>

<url-pattern>/setup/\*</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>OpenTenantGateFilter</filter-name>

<url-pattern>/bootstrap/\*</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>OpenTenantGateFilter</filter-name>

<url-pattern>/johnson/\*</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>OpenTenantGateFilter</filter-name>

<url-pattern>/errors.jsp</url-pattern>

</filter-mapping>

<!-- Plugins 2.5 filter changes -->

<filter-mapping>

<filter-name>filter-plugin-dispatcher-after-encoding-request</filter-name>

<url-pattern>/\*</url-pattern>

<dispatcher>REQUEST</dispatcher>

</filter-mapping>

<filter-mapping>

<filter-name>filter-plugin-dispatcher-after-encoding-forward</filter-name>

<url-pattern>/\*</url-pattern>

<dispatcher>FORWARD</dispatcher>

</filter-mapping>

<filter-mapping>

<filter-name>filter-plugin-dispatcher-after-encoding-include</filter-name>

<url-pattern>/\*</url-pattern>

<dispatcher>INCLUDE</dispatcher>

</filter-mapping>

<filter-mapping>

<filter-name>filter-plugin-dispatcher-after-encoding-error</filter-name>

<url-pattern>/\*</url-pattern>

<dispatcher>ERROR</dispatcher>

</filter-mapping>

<!-- End plugins 2.5 filter changes -->

<filter-mapping>

<filter-name>caching</filter-name>

<url-pattern>/\*</url-pattern>

</filter-mapping>

<!-- uncomment this mapping in order to log page views to the access log, see log4j.properties also -->

<!--

<filter-mapping>

<filter-name>AccessLogFilter</filter-name>

<url-pattern>/display/\*</url-pattern>

</filter-mapping>

-->

<filter-mapping>

<filter-name>legacyRemoteApiEventPublishingFilter</filter-name>

<url-pattern>/plugins/servlet/soap-axis1/\*</url-pattern>

<url-pattern>/rpc/\*</url-pattern>

<url-pattern>/rest/prototype/\*</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>RequestTimeFilter</filter-name>

<url-pattern>\*.action</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>RequestTimeFilter</filter-name>

<url-pattern>\*.vm</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>RequestTimeFilter</filter-name>

<url-pattern>/display/\*</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>RequestTimeFilter</filter-name>

<url-pattern>/plugins/servlet/\*</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>RequestTimeFilter</filter-name>

<url-pattern>/download/\*</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>RequestTimeFilter</filter-name>

<url-pattern>/label/\*</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>RequestTimeFilter</filter-name>

<url-pattern>/s/\*</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>RequestTimeFilter</filter-name>

<url-pattern>/questions/\*</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>profiling</filter-name>

<url-pattern>\*.action</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>profiling</filter-name>

<url-pattern>\*.vm</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>profiling</filter-name>

<url-pattern>/display/\*</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>profiling</filter-name>

<url-pattern>/plugins/servlet/\*</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>profiling</filter-name>

<url-pattern>/download/\*</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>profiling</filter-name>

<url-pattern>/label/\*</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>profiling</filter-name>

<url-pattern>/rest/\*</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>profiling</filter-name>

<url-pattern>/rpc/\*</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>profiling</filter-name>

<url-pattern>/s/\*</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>thread-local-error-collection</filter-name>

<url-pattern>\*.action</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>thread-local-error-collection</filter-name>

<url-pattern>\*.vm</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>thread-local-error-collection</filter-name>

<url-pattern>/display/\*</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>thread-local-error-collection</filter-name>

<url-pattern>/plugins/servlet/\*</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>thread-local-error-collection</filter-name>

<url-pattern>/download/\*</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>thread-local-error-collection</filter-name>

<url-pattern>/label/\*</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>thread-local-error-collection</filter-name>

<url-pattern>/s/\*</url-pattern>

</filter-mapping>

<!--filter-mapping>

<filter-name>sessioninview</filter-name>

<url-pattern>/rpc/\*</url-pattern>

</filter-mapping-->

<filter-mapping>

<filter-name>sessioninview</filter-name>

<url-pattern>\*.action</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>sessioninview</filter-name>

<url-pattern>\*.vm</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>sessioninview</filter-name>

<url-pattern>/display/\*</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>sessioninview</filter-name>

<url-pattern>/plugins/servlet/\*</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>sessioninview</filter-name>

<url-pattern>/label/\*</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>sessioninview</filter-name>

<url-pattern>/s/\*</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>sessioninview</filter-name>

<url-pattern>/exportword</url-pattern>

</filter-mapping>

<!-- Wrap the prototype Confluence REST plugin in a transaction, as the REST plugin type does not have effective support

for transactions yet. Hopefully non-prototype REST implementations will support transactions properly so we don't

have to extend this hack to production code -->

<filter-mapping>

<filter-name>sessioninview</filter-name>

<url-pattern>/rest/\*</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>ClusterHeaderFilter</filter-name>

<url-pattern>/\*</url-pattern>

</filter-mapping>

<!-- Plugins 2.5 filter changes -->

<filter-mapping>

<filter-name>filter-plugin-dispatcher-before-login-request</filter-name>

<url-pattern>/\*</url-pattern>

<dispatcher>REQUEST</dispatcher>

</filter-mapping>

<filter-mapping>

<filter-name>filter-plugin-dispatcher-before-login-forward</filter-name>

<url-pattern>/\*</url-pattern>

<dispatcher>FORWARD</dispatcher>

</filter-mapping>

<filter-mapping>

<filter-name>filter-plugin-dispatcher-before-login-include</filter-name>

<url-pattern>/\*</url-pattern>

<dispatcher>INCLUDE</dispatcher>

</filter-mapping>

<filter-mapping>

<filter-name>filter-plugin-dispatcher-before-login-error</filter-name>

<url-pattern>/\*</url-pattern>

<dispatcher>ERROR</dispatcher>

</filter-mapping>

<!-- End plugins 2.5 filter changes -->

<!-- Limit authentication metrics to just the /display path, to avoid false hits on AJAX background requests -->

<filter-mapping>

<filter-name>authenticator-metrics</filter-name>

<url-pattern>/display/\*</url-pattern>

<dispatcher>REQUEST</dispatcher>

</filter-mapping>

<filter-mapping>

<filter-name>login</filter-name>

<url-pattern>/\*</url-pattern>

<dispatcher>FORWARD</dispatcher>

<dispatcher>REQUEST</dispatcher>

<dispatcher>ERROR</dispatcher>

</filter-mapping>

<!-- This must come after the login filter -->

<filter-mapping>

<filter-name>trustedapp</filter-name>

<url-pattern>/\*</url-pattern>

</filter-mapping>

<!-- SecurityFilter can use the ThreadLocalCache, so we initialise it before -->

<filter-mapping>

<filter-name>threadLocalCache</filter-name>

<url-pattern>\*.action</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>threadLocalCache</filter-name>

<url-pattern>\*.vm</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>threadLocalCache</filter-name>

<url-pattern>\*.jsp</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>threadLocalCache</filter-name>

<url-pattern>/display/\*</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>threadLocalCache</filter-name>

<url-pattern>/label/\*</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>threadLocalCache</filter-name>

<url-pattern>/exportword</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>threadLocalCache</filter-name>

<url-pattern>/rpc/\*</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>threadLocalCache</filter-name>

<url-pattern>/s/\*</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>threadLocalCache</filter-name>

<url-pattern>/rest/\*</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>threadLocalCache</filter-name>

<url-pattern>/plugins/servlet/\*</url-pattern>

</filter-mapping>

<!-- Downloads use the ThreadLocalPermissionCache -->

<filter-mapping>

<filter-name>threadLocalCache</filter-name>

<url-pattern>/download/\*</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>security</filter-name>

<url-pattern>/\*</url-pattern>

<dispatcher>FORWARD</dispatcher>

<dispatcher>REQUEST</dispatcher>

<dispatcher>ERROR</dispatcher>

</filter-mapping>

<!-- Must go after seraph -->

<filter-mapping>

<filter-name>timeout</filter-name>

<url-pattern>/\*</url-pattern>

</filter-mapping>

<!-- Must go after seraph -->

<filter-mapping>

<filter-name>userthreadlocal</filter-name>

<url-pattern>\*.action</url-pattern>

<dispatcher>FORWARD</dispatcher>

<dispatcher>REQUEST</dispatcher>

<dispatcher>ERROR</dispatcher>

</filter-mapping>

<filter-mapping>

<filter-name>userthreadlocal</filter-name>

<url-pattern>\*.vm</url-pattern>

<dispatcher>FORWARD</dispatcher>

<dispatcher>REQUEST</dispatcher>

</filter-mapping>

<filter-mapping>

<filter-name>userthreadlocal</filter-name>

<url-pattern>/display/\*</url-pattern>

<dispatcher>FORWARD</dispatcher>

<dispatcher>REQUEST</dispatcher>

</filter-mapping>

<filter-mapping>

<filter-name>userthreadlocal</filter-name>

<url-pattern>/label/\*</url-pattern>

<dispatcher>FORWARD</dispatcher>

<dispatcher>REQUEST</dispatcher>

</filter-mapping>

<filter-mapping>

<filter-name>userthreadlocal</filter-name>

<url-pattern>/exportword</url-pattern>

<dispatcher>FORWARD</dispatcher>

<dispatcher>REQUEST</dispatcher>

</filter-mapping>

<filter-mapping>

<filter-name>userthreadlocal</filter-name>

<url-pattern>/s/\*</url-pattern>

<dispatcher>FORWARD</dispatcher>

<dispatcher>REQUEST</dispatcher>

</filter-mapping>

<filter-mapping>

<filter-name>maueventfilter</filter-name>

<url-pattern>/\*</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>usernameheader</filter-name>

<url-pattern>/\*</url-pattern>

<dispatcher>FORWARD</dispatcher>

<dispatcher>REQUEST</dispatcher>

</filter-mapping>

<filter-mapping>

<filter-name>UserLoggingContextFilter</filter-name>

<url-pattern>/\*</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>servletcontextthreadlocal</filter-name>

<url-pattern>\*.action</url-pattern>

<dispatcher>FORWARD</dispatcher>

<dispatcher>REQUEST</dispatcher>

</filter-mapping>

<filter-mapping>

<filter-name>servletcontextthreadlocal</filter-name>

<url-pattern>\*.vm</url-pattern>

<dispatcher>FORWARD</dispatcher>

<dispatcher>REQUEST</dispatcher>

</filter-mapping>

<filter-mapping>

<filter-name>servletcontextthreadlocal</filter-name>

<url-pattern>/display/\*</url-pattern>

<dispatcher>FORWARD</dispatcher>

<dispatcher>REQUEST</dispatcher>

</filter-mapping>

<filter-mapping>

<filter-name>servletcontextthreadlocal</filter-name>

<url-pattern>/label/\*</url-pattern>

<dispatcher>FORWARD</dispatcher>

<dispatcher>REQUEST</dispatcher>

</filter-mapping>

<filter-mapping>

<filter-name>servletcontextthreadlocal</filter-name>

<url-pattern>/rpc/\*</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>servletcontextthreadlocal</filter-name>

<url-pattern>/plugins/\*</url-pattern>

<dispatcher>FORWARD</dispatcher>

<dispatcher>REQUEST</dispatcher>

</filter-mapping>

<filter-mapping>

<filter-name>servletcontextthreadlocal</filter-name>

<url-pattern>/s/\*</url-pattern>

<dispatcher>FORWARD</dispatcher>

<dispatcher>REQUEST</dispatcher>

</filter-mapping>

<filter-mapping>

<filter-name>servletcontextthreadlocal</filter-name>

<url-pattern>/download/\*</url-pattern>

<dispatcher>FORWARD</dispatcher>

<dispatcher>REQUEST</dispatcher>

</filter-mapping>

<filter-mapping>

<filter-name>servletcontextthreadlocal</filter-name>

<url-pattern>/rest/\*</url-pattern>

<dispatcher>FORWARD</dispatcher>

<dispatcher>REQUEST</dispatcher>

</filter-mapping>

<filter-mapping>

<filter-name>transactionalCacheFactoryCleanupFilter</filter-name>

<url-pattern>\*.action</url-pattern>

<dispatcher>REQUEST</dispatcher>

</filter-mapping>

<filter-mapping>

<filter-name>transactionalCacheFactoryCleanupFilter</filter-name>

<url-pattern>\*.vm</url-pattern>

<dispatcher>REQUEST</dispatcher>

</filter-mapping>

<filter-mapping>

<filter-name>transactionalCacheFactoryCleanupFilter</filter-name>

<url-pattern>/display/\*</url-pattern>

<dispatcher>REQUEST</dispatcher>

</filter-mapping>

<filter-mapping>

<filter-name>transactionalCacheFactoryCleanupFilter</filter-name>

<url-pattern>/label/\*</url-pattern>

<dispatcher>REQUEST</dispatcher>

</filter-mapping>

<filter-mapping>

<filter-name>transactionalCacheFactoryCleanupFilter</filter-name>

<url-pattern>/rpc/\*</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>transactionalCacheFactoryCleanupFilter</filter-name>

<url-pattern>/s/\*</url-pattern>

<dispatcher>REQUEST</dispatcher>

</filter-mapping>

<filter-mapping>

<filter-name>transactionalCacheFactoryCleanupFilter</filter-name>

<url-pattern>/download/\*</url-pattern>

<dispatcher>REQUEST</dispatcher>

</filter-mapping>

<filter-mapping>

<filter-name>transactionalCacheFactoryCleanupFilter</filter-name>

<url-pattern>/rest/\*</url-pattern>

<dispatcher>REQUEST</dispatcher>

</filter-mapping>

<filter-mapping>

<filter-name>jmx</filter-name>

<url-pattern>\*.action</url-pattern>

</filter-mapping>

<filter-mapping>

<filter-name>jmx</filter-name>

<url-pattern>/display/\*</url-pattern>

</filter-mapping>

<!-- The UrlRewriteFilter also forwards to other paths and we want to make sure the /s/\* filter mappings run

before this one to set up caches and other things -->

<filter-mapping>

<filter-name>UrlRewriteFilter</filter-name>

<url-pattern>/s/\*</url-pattern>

</filter-mapping>

<!-- CONFDEV-14301: This mapping should only be used in order to gracefully deprecate icon URLs -->

<filter-mapping>

<filter-name>UrlRewriteFilter</filter-name>

<url-pattern>/images/icons/\*</url-pattern>

</filter-mapping>

<!-- Plugins 2.5 filter changes -->

<filter-mapping>

<filter-name>filter-plugin-dispatcher-before-decoration-request</filter-name>

<url-pattern>/\*</url-pattern>

<dispatcher>REQUEST</dispatcher>

</filter-mapping>

<filter-mapping>

<filter-name>filter-plugin-dispatcher-before-decoration-forward</filter-name>

<url-pattern>/\*</url-pattern>

<dispatcher>FORWARD</dispatcher>

</filter-mapping>

<filter-mapping>

<filter-name>filter-plugin-dispatcher-before-decoration-include</filter-name>

<url-pattern>/\*</url-pattern>

<dispatcher>INCLUDE</dispatcher>

</filter-mapping>

<filter-mapping>

<filter-name>filter-plugin-dispatcher-before-decoration-error</filter-name>

<url-pattern>/\*</url-pattern>

<dispatcher>ERROR</dispatcher>

</filter-mapping>

<!-- End plugins 2.5 filter changes -->

<filter-mapping>

<filter-name>sitemesh</filter-name>

<url-pattern>/\*</url-pattern>

<dispatcher>REQUEST</dispatcher>

</filter-mapping>

<filter-mapping>

<filter-name>sitemesh-error</filter-name>

<url-pattern>/\*</url-pattern>

<dispatcher>ERROR</dispatcher>

</filter-mapping>

<!-- Needs to run before the sitemesh filter works with the response but after the target servlet/resource was executed -->

<filter-mapping>

<filter-name>messagesDecoratorFilter</filter-name>

<url-pattern>\*.action</url-pattern>

<dispatcher>REQUEST</dispatcher>

<dispatcher>FORWARD</dispatcher>

</filter-mapping>

<filter-mapping>

<filter-name>messagesDecoratorFilter</filter-name>

<url-pattern>\*.vm</url-pattern>

<dispatcher>REQUEST</dispatcher>

<dispatcher>FORWARD</dispatcher>

</filter-mapping>

<filter-mapping>

<filter-name>messagesDecoratorFilter</filter-name>

<url-pattern>/display/\*</url-pattern>

<dispatcher>REQUEST</dispatcher>

<dispatcher>FORWARD</dispatcher>

</filter-mapping>

<filter-mapping>

<filter-name>messagesDecoratorFilter</filter-name>

<url-pattern>/plugins/servlet/\*</url-pattern>

<dispatcher>REQUEST</dispatcher>

<dispatcher>FORWARD</dispatcher>

</filter-mapping>

<filter-mapping>

<filter-name>messagesDecoratorFilter</filter-name>

<url-pattern>/label/\*</url-pattern>

<dispatcher>REQUEST</dispatcher>

<dispatcher>FORWARD</dispatcher>

</filter-mapping>

<filter-mapping>

<filter-name>expires-one-hour</filter-name>

<url-pattern>\*.js</url-pattern>

</filter-mapping>

<!--<filter-mapping>

<filter-name>expires-one-hour</filter-name>

<url-pattern>\*.css</url-pattern>

</filter-mapping>-->

<!-- Plugins 2.5 filter changes -->

<!-- the following plugin filter should be the last one in the chain -->

<filter-mapping>

<filter-name>filter-plugin-dispatcher-before-dispatch-request</filter-name>

<url-pattern>/\*</url-pattern>

<dispatcher>REQUEST</dispatcher>

</filter-mapping>

<filter-mapping>

<filter-name>filter-plugin-dispatcher-before-dispatch-forward</filter-name>

<url-pattern>/\*</url-pattern>

<dispatcher>FORWARD</dispatcher>

</filter-mapping>

<filter-mapping>

<filter-name>filter-plugin-dispatcher-before-dispatch-include</filter-name>

<url-pattern>/\*</url-pattern>

<dispatcher>INCLUDE</dispatcher>

</filter-mapping>

<filter-mapping>

<filter-name>filter-plugin-dispatcher-before-dispatch-error</filter-name>

<url-pattern>/\*</url-pattern>

<dispatcher>ERROR</dispatcher>

</filter-mapping>

<!-- End plugins 2.5 filter changes -->

<filter-mapping>

<filter-name>debug-after-request</filter-name>

<url-pattern>/\*</url-pattern>

<dispatcher>REQUEST</dispatcher>

</filter-mapping>

<filter-mapping>

<filter-name>debug-after-include</filter-name>

<url-pattern>/\*</url-pattern>

<dispatcher>INCLUDE</dispatcher>

</filter-mapping>

<filter-mapping>

<filter-name>debug-after-forward</filter-name>

<url-pattern>/\*</url-pattern>

<dispatcher>FORWARD</dispatcher>

</filter-mapping>

<filter-mapping>

<filter-name>debug-after-error</filter-name>

<url-pattern>/\*</url-pattern>

<dispatcher>ERROR</dispatcher>

</filter-mapping>

<!-- ============================================================ -->

<!-- Servlet Context Listeners (Executed on app startup/shutdown) -->

<!-- ============================================================ -->

<!-- Initialize Johnson -->

<listener>

<listener-class>com.atlassian.johnson.context.JohnsonContextListener</listener-class>

</listener>

<!-- Cleans up JavaBeans introspection caches on app shutdown, so that the classes and classloaders can be

garbage-collected properly -->

<listener>

<listener-class>org.springframework.web.util.IntrospectorCleanupListener</listener-class>

</listener>

<!-- Load initial minimal configuration and bootstrap the application ready for setup -->

<listener>

<listener-class>com.atlassian.confluence.setup.ConfluenceConfigurationListener</listener-class>

</listener>

<!-- Bring up the rest of the application if it is already set up -->

<listener>

<listener-class>com.atlassian.confluence.setup.ConfluenceBootstrappedContextLoaderListener</listener-class>

</listener>

<!-- Check for the confluence.i18n.reloadbundles system property. If set, do not cache the localized

.properties files. -->

<listener>

<listener-class>com.atlassian.confluence.languages.ReloadBundlesContextListener</listener-class>

</listener>

<listener>

<listener-class>com.atlassian.confluence.setup.ValidLicenseContextListener</listener-class>

</listener>

<!-- Bring plugin system up, if the database is configured. Plugins must be up before we can run the lifecycle modules. -->

<listener>

<listener-class>com.atlassian.confluence.plugin.PluginFrameworkContextListener</listener-class>

</listener>

<!-- Check if the system is under recovery mode and create recovery\_admin user if not existing -->

<listener>

<listener-class>com.atlassian.confluence.impl.security.recovery.RecoveryContextListener</listener-class>

</listener>

<!-- Perform remaining configured lifecycle events, if the application is set up -->

<listener>

<listener-class>com.atlassian.config.lifecycle.LifecycleServletContextListener</listener-class>

</listener>

<!-- ===================================== -->

<!-- Other (non servlet-context) listeners -->

<!-- ===================================== -->

<listener>

<listener-class>com.atlassian.confluence.user.listeners.UserSessionExpiryListener</listener-class>

</listener>

<listener>

<listener-class>com.atlassian.confluence.util.http.ConfluenceAttributeListener</listener-class>

</listener>

<!-- ======== -->

<!-- Servlets -->

<!-- ======== -->

<servlet>

<servlet-name>action</servlet-name>

<servlet-class>com.atlassian.confluence.servlet.ConfluenceServletDispatcher</servlet-class>

<load-on-startup>1</load-on-startup>

</servlet>

<servlet>

<servlet-name>velocity</servlet-name>

<servlet-class>com.atlassian.confluence.servlet.ConfluenceVelocityServlet</servlet-class>

<load-on-startup>2</load-on-startup>

</servlet>

<servlet>

<servlet-name>simple-display</servlet-name>

<servlet-class>com.atlassian.confluence.servlet.SpringManagedServlet</servlet-class>

<init-param>

<param-name>springComponentName</param-name>

<param-value>simpleDisplayServlet</param-value>

</init-param>

<load-on-startup>3</load-on-startup>

</servlet>

<servlet>

<servlet-name>tiny-url</servlet-name>

<servlet-class>com.atlassian.confluence.servlet.TinyUrlServlet</servlet-class>

<load-on-startup>3</load-on-startup>

</servlet>

<servlet>

<servlet-name>file-server</servlet-name>

<servlet-class>com.atlassian.confluence.servlet.FileServerServlet</servlet-class>

<load-on-startup>4</load-on-startup>

</servlet>

<servlet>

<servlet-name>status-servlet</servlet-name>

<servlet-class>com.atlassian.confluence.servlet.ApplicationStatusServlet</servlet-class>

<load-on-startup>5</load-on-startup>

</servlet>

<servlet>

<servlet-name>xmlrpc</servlet-name>

<servlet-class>com.atlassian.confluence.servlet.SpringManagedServlet</servlet-class>

<init-param>

<param-name>springComponentName</param-name>

<param-value>xmlRpcServer</param-value>

</init-param>

<load-on-startup>6</load-on-startup>

</servlet>

<servlet>

<servlet-name>servlet-module-container-servlet</servlet-name>

<servlet-class>com.atlassian.confluence.servlet.ServletModuleContainerServlet</servlet-class>

<load-on-startup>9</load-on-startup>

</servlet>

<servlet>

<servlet-name>css</servlet-name>

<servlet-class>com.atlassian.confluence.servlet.CssServlet</servlet-class>

<load-on-startup>10</load-on-startup>

</servlet>

<!-- Keep this servlet as the last to load -->

<servlet>

<servlet-name>final-servlet</servlet-name>

<servlet-class>com.atlassian.confluence.servlet.ReadyToServeServlet</servlet-class>

<load-on-startup>100</load-on-startup>

</servlet>

<servlet>

<servlet-name>labels</servlet-name>

<servlet-class>com.atlassian.confluence.servlet.LabelServlet</servlet-class>

</servlet>

<servlet>

<servlet-name>jcaptcha</servlet-name>

<servlet-class>com.atlassian.confluence.servlet.ImageCaptchaServlet</servlet-class>

<load-on-startup>0</load-on-startup>

</servlet>

<servlet>

<servlet-name>exportword</servlet-name>

<servlet-class>com.atlassian.confluence.servlet.SpringManagedServlet</servlet-class>

<init-param>

<param-name>springComponentName</param-name>

<param-value>exportWordPageServer</param-value>

</init-param>

</servlet>

<!-- Dummy servlet for CONF-7953. Used for mapping URLs which have no target servlet but need to be filtered -->

<servlet>

<servlet-name>noop</servlet-name>

<servlet-class>com.atlassian.confluence.servlet.ConfluenceNoOpServlet</servlet-class>

<load-on-startup>0</load-on-startup>

</servlet>

<servlet>

<servlet-name>johnson-analytics-servlet</servlet-name>

<servlet-class>com.atlassian.confluence.servlet.JohnsonAnalyticsServlet</servlet-class>

</servlet>

<servlet>

<servlet-name>johnson-data-servlet</servlet-name>

<servlet-class>com.atlassian.confluence.servlet.JohnsonDataServlet</servlet-class>

</servlet>

<servlet>

<servlet-name>johnson-dismiss-events-servlet</servlet-name>

<servlet-class>com.atlassian.confluence.servlet.JohnsonDismissEventsServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>action</servlet-name>

<url-pattern>\*.action</url-pattern>

</servlet-mapping>

<!--

we pretty much have to map all CSS files to the action servlet, as a result

of a fun collaboration of an IE bug, and the short-sightedness of the servlet

spec.

-->

<servlet-mapping>

<servlet-name>css</servlet-name>

<url-pattern>\*.css</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>velocity</servlet-name>

<url-pattern>\*.vm</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>simple-display</servlet-name>

<url-pattern>/display/\*</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>tiny-url</servlet-name>

<url-pattern>/x/\*</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>file-server</servlet-name>

<url-pattern>/download/\*</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>status-servlet</servlet-name>

<url-pattern>/status</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>xmlrpc</servlet-name>

<url-pattern>/rpc/xmlrpc</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>servlet-module-container-servlet</servlet-name>

<url-pattern>/plugins/servlet/\*</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>labels</servlet-name>

<url-pattern>/label/\*</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>jcaptcha</servlet-name>

<url-pattern>/jcaptcha/\*</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>exportword</servlet-name>

<url-pattern>/exportword</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>noop</servlet-name>

<url-pattern>/s/\*</url-pattern>

</servlet-mapping>

<!--

Noop filter mapping for the trusted app certificate which is serviced exclusively by the Seraph trusted app filter.

This servlet mapping is necessary for Websphere 6.1 which still likes to forward the request to a servlet even if

the filter chain is not fully followed.

-->

<servlet-mapping>

<servlet-name>noop</servlet-name>

<url-pattern>/admin/appTrustCertificate</url-pattern>

</servlet-mapping>

<!--

As the REST module is implemented using only a filter this noop servlet is required (CONF-17578)

-->

<servlet-mapping>

<servlet-name>noop</servlet-name>

<url-pattern>/rest/\*</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>johnson-analytics-servlet</servlet-name>

<url-pattern>/johnson/analytics/\*</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>johnson-data-servlet</servlet-name>

<url-pattern>/johnson/data</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>johnson-dismiss-events-servlet</servlet-name>

<url-pattern>/johnson/events/dismiss</url-pattern>

</servlet-mapping>

<session-config>

<session-timeout>60</session-timeout>

<tracking-mode>COOKIE</tracking-mode>

</session-config>

<welcome-file-list>

<welcome-file>default.jsp</welcome-file>

<welcome-file>index.action</welcome-file>

</welcome-file-list>

<!-- redirect all 500 errors to confluence error page -->

<error-page>

<error-code>500</error-code>

<location>/500page.jsp</location>

</error-page>

<error-page>

<error-code>404</error-code>

<location>/fourohfour.action</location>

</error-page>

<error-page>

<exception-type>com.atlassian.sal.api.permission.NotAuthenticatedException</exception-type>

<location>/login.action</location>

</error-page>

<error-page>

<exception-type>com.atlassian.sal.api.permission.AuthorisationException</exception-type>

<location>/notpermitted.action</location>

</error-page>

</web-app>