

Logging Project Path Forward Proposal



The following proposal was made by Mike Elliott, former Logging PTL.

« Review what it is really used by the ONAP Projects, move the repositories under another umbrella ONAP Project's umbrella and decommission the rest »

Action Plan to be reviewed/approved by ONAP TSC

#1 E-mail sent to the ONAP Community – Final Call to support « Logging » Project – **May 29th**

If not positive feedback then implement Mike Elliott's proposal

Today

#2 Repositories currently used by ONAP projects (TBC by PTLs)

<https://gerrit.onap.org/r/gitweb?p=logging-analytics.git;a=tree;f=reference/logging-filter;h=78d5995fd54b580fbda4245ccfdee26504a45a30;hb=refs/heads/master>

<https://gerrit.onap.org/r/gitweb?p=logging-analytics.git;a=tree;f=reference/logging-slf4j;h=aac6f850373892b1da131405521d527751588f8e;hb=refs/heads/master>

#3 Where are the logs (configurable in Helm Chart)? **`/var/log/onap`**

#4 Identify the ONAP Project Umbrella: OOM

#5 Commitments to support the repository: AT&T (former Logging committer), anybody else?

#6 Move Logging to Project State: Archived following the [Termination Review process](#)

Next Step

#7 As part of our Cloud Native transformation, setup task force to analyse how we can also offer the 'sdtout' as an option

Information about the « Logging » repositories



```
1 # Logging-Filter
2
3 The base project is meant to have very minimal dependencies for maximum re-use. The spring artifact is provided for any clients already using spring.
4
5 ## Current Layout
6 - logging-filter (houses the parent pom)
7 - **base** this artifact should only depend on javax.servlet-api, javax.ws.rs-api, slf4j-api and org.onap.logging-analytics:logging-slf4j
8 - **spring** this artifact depends on base as well as spring.
9
10 ## Design Principles
11 - minimize dependencies for maximum re-use
12 - code to APIs jaxrs vs jersey, slf4j vs logback, etc...
13 - no application specific code in this library
14 - write extensible code so applications can add on application specific code
15 - target Casablanca logging spec
16
17 ## Code Formatting
18 ONAP code formatting standards are validated on builds. If you find a build failing due to a formatting issue, format your code by running `mvn process-sources -P format`
19
20
21 # README - slf4j-reference
22
23 This project gives an example of ONAP-compliant logging using SLF4J logging.
24
25 ## Adapter
26
27 In ``org.onap.logging.ref.slf4j``, there are TWO classes:
28 1. ``org.onap.logging.ref.slf4j.ONAPLogConstants``, providing declarations of standard ONAP Markers, MDCs and HTTP headers.
29 2. ``org.onap.logging.ref.slf4j.ONAPLogAdapter``, providing a lightweight, compliant implementation of the ONAP logging spec.
30
31 The adapter provides:
32 1. A loosely-coupled SLF4j logging wrapper:
33 * To be used for logging ONAP ``entry``, ``exit`` and ``invoke`` behavior.
34 * Devolving all *application* logging to the component, via the regular SLF4J ``Logger`` facade.
35 2. Customization options:
36 * *Cheap*, by way of bean properties. This is suited to most Use Cases.
37 * *Sophisticated*:
38 * By OPTIONALLY implementing one of a number of adapters:
39 * ``RequestAdapter`` to read incoming headers.
40 * ``ServiceDescriptor`` for reporting attributes of the current service.
41 * ``ResponseDescriptor`` for reporting outcomes.
42 * ``RequestBuilder`` for setting southbound request headers.
43 * By OPTIONALLY overriding methods like ``ONAPLogAdapter#setMDCs(RequestAdapter)``.
44
45 Note that:
46 * The adapter implementation uses static inner classes in order to fit in a single source file. This was an objective.
47
48 The purpose of the WAR is to demonstrate minimalist ONAP-compliant logging for web components, but a secondary purpose is to demonstrate that the call graph can be generated for a (mostly) representative set of interacting REST services.
49
50 ## Configuration
51
52 Note that the bundled SLF4J configuration isn't meant to be normative. Close enough as makes no difference for
53 the purposes of tests, but be aware that it might not always be up-to-date.
54
55 ## Tests
56
57 TestNG tests for contract + output of SLF4J reference impl.
```