Hackathon in Berlin - September 2023

DONE

Location: **diva-e, Berlin Südkreuz**
Date: **September 28th, 2023**
Time: **10:00 - 16:45**

Participants

- Robert Munteanu
- Stefan Seifert
- Konrad Windszus
- Oliver Lietz
- Radu Cotescu
- Karl Pauls
- Joerg Hoh
- Carsten Ziegeler
- Dominik Süß

Proposed topics

Topic	Summary	Participants	Proposed duration
Switch to Jakarta APIs	**SLING-10000** - Getting issue details... STATUS		
Is Sling becoming too fat?	In e.g. Sling XSS there is a considerable increase in dependencies.		
Is maintaining proper features (OSGi, Karaf) worth the effort?	In 936eb9f a bundle was added to feature models-jacksonexporter for XSS. Could we use dependencies to create and manage proper features? Is it supported by OSGi Features? > *Features may have dependencies on other Features.* Tie-in discussion: how do we structure our features? We currently have: ▪ library : jackson ▪ function: validation, maintenance, scripting ▪ deployment: docker There is also a question of whether the healthcheck webconsole bundle belongs in the healthcheck or the webconsole feature. Or maybe in a healthcheck-webconsole feature?		
Adopt SLF4J 2.x	**SLING-11906** - Getting issue details... STATUS		
Enforcing CI checks for pull requests	Classify modules and enfore specific CI checks, based on module maturity/risk profile.		

Migrate Sling Models /CAConfig Integration Tests to Sling 12/Feature Model	Integration tests are currently broken or use very outdated Sling Starter. Goal would be to switch to Sling 12+Feature Module and move the integration tests in the actual codebase of the implementation classes, no separate codebase for ITs. see also discussion in https://lists.apache.org/thread/fwwnb2hyqf2t2ko4pfmhhxd7wrhwmbh6 WIP in https://github.com/apache/sling-org-apache-sling-models-impl/pull/46		
Required Java Versions for Building and Running	Bnd 7.x requires Java 17+ (https://github.com/bndtools/bnd/wiki/Changes-in-7.0.0) Drop Java 8 x support at run time (**SLING-11842** - Getting issue details... STATUS) Drop also Java 11 support at run time?		
Changing poms to reference GitHub (url only)	All our POMs reference gitbox for the 'browse' URL, e.g. https://gitbox.apache.org/repos/asf?p=sling-org-apache-sling-engine.git . These URLs currently redirect to github. The 'browse URL' is sometimes used by tooling to point people to repositories and using GitHub directly might be nicer. Also tooling like renovate uses the github.com URLs to extract changelogs for bundle updates.		
Create 'apache/sling' entry point repository on GitHub	A long time ago (6 years?) we set up a redirect from apache/sling to https://github.com/apache/sling-old-svn-mirror as we moved away to invidual modules. At this point this repository has served its usefulness and we can probably stop worrying about stale links and create a nice, welcoming, entry point repository at https://github.com/apache/sling that is better serving new users.		

Discussed Topics

List of topics discussed during Sling Committer Round Table (Sept. 26th 2023) and Sling Hackathon (Sept. 28th 2023), most of them marked with a Community Member who plans to bring this discussion to the mailing list or make further proposals.

1) Jakarta Servlet API

- With Jakarta Servlet API 5 all package names changed to Jakarta Namespace. In Jakrarta Servlet API 6 all deprecated methods from previous versions are removed.
- Felix provides two whiteboards for old and new Servlet API, both can be used in parallel in one context
- How to deal with those changes in Sling? Sling has direct dependencies to the Servlet API as part of it's APIs, so it does not only affect the implementation.
  - option 1: provide new API interfaces based on Jakarta - parallel to the old ones with side-by-side packages
    - maybe this should focus on servlet 6 (with removed deprecated stuff), to not have two migration steps for the users
    - it's also an opportunity to clean up our own API around the Servlet API
  - option 2: existing servlet-dependant apis switch to jakarata without changing the package name, fix incompatibilites e.g. by adding missing methods, may become messy
  - option 3: do nothing now and wait until this switch is actually becoming a problem
- most feasible option seems to be option 1 - but it's a lot of work and risk to break stuff

Carsten Ziegeler will start the discussion - dev@sling - [RT] Path to Jakarta Servlet Specification

2) Java Version Support

- Latest **bnd version** will support only java 17 on build time
- question: should we use build only with java 17 but with source release for java 11?
- otherwise we're stuck with old/unmaintained bnd version
- running integration test with java 11 gets more complicated
- we also want to add **Java 21** as default build for our CIs (it's LTS and GA now) tracked in **SLING-12056** - Getting issue details... STATUS

Konrad Windszus

3) JIRA Cleanup

- We have tons of issues open JIRA issues, lot of them not touched for years.
- Carsten recently closed a couple ones that where older then ~8 years
- Robert will look through a list of open tickets assigned to him

4) Unify ways of doing ITs in Sling? / Bringing ITs to implementation repo

- Currently we have quite a lot of different approaches in Sling Modules to do the ITs
- Roughly the can be categorized in two types of tests: ITs against a real Sling instance via HTTP, and ITs using Paxexam. Those both categories use a quite different approach, and it's fine to have both depending on the use cases. In both areas, there are modules using outdated libraries /approaches.
- Especially for the ITs running against Sling instance we have a couple ones very outdated using Provisioning Model and stored in a separate repository (especially Sling Models, Context-Aware Config). Those should be migrated to use Feature Model, run against the latest Sling version, and should be moved to the implementation repository. Otherwise there are often broken and difficult to maintain.

Konrad Windszus Stefan Seifert to bring the ITs for models and context-aware configuration in an acceptable shape

5) SLF4J 2

- SLF4J 2 is there and some 3rdparty libs start using it exclusively. Although the main API itself is properly maintained and versioned and an be deployed in a backward compatible way, that's not the case for additional helper APIs, which are also used in some places.
- Konrad discussed this a bit with the SLF4J project itself but not with a successful result yet (https://github.com/qos-ch/slf4j/pull/358)

Konrad Windszus will come up with a lightweight first proposal

6) Generate Karaf features out of OSGi features

- currently, it's a tedious and mostly manual process to derive the Karaf features from the Sling OSGi features (maintained by Olli). at the same time, the Karaf features reflect the granularity much better then the current Sling features, and the also declare dependencies between the features, which is currently not possible for the OSGi features.
- the goals is to restructure the Sling feature in a way to reflect the granularity of Karaf features, and additional metadata in them that represent the dependencies between the features (this metadata is only for the Karaf translation)
- with this, it should be possible to generate the Karaf features automatically from the OSGi features and have a single truth for those

Carsten Ziegeler Oliver Lietz

7) Better Search for Sling Website

- Bertrand proposed to provide a better search for the Sling Websites based on the generated Site content using https://pagefind.app/ (also used in other Apache websites)
- PR proposal: https://github.com/apache/sling-site/pull/133
- This search does not include the content from the READMEs in GitHub, some modules have their documentation only there (we discussed this a bit as well) - in the meantime this has been clarified in the "300 modules" section of https://sling.apache.org/documentation.html

Bertrand Delacretaz

8) Changing POM URL References to GitHub

- Currently, the "Browsing URL" in the POMs is pointing to GitBox. Those URLs are used by various tooling and humans.
- We should switch it directly to GitHub to easy the discovery process
- The Source Code URLs used for Release Process stick with GitBox

Robert Munteanu - **SLING-12094** - Getting issue details... STATUS

9) Create 'apache/sling" entry point on GitHub

- Currently, going to github.com/apache/sling results in a suboptimal user experience, pointing to an obsolete code repository
- Proposal to set up a new "sling" repository mainly consisting of a nice README pointing to the further information and steps

Robert Munteanu / **SLING-12095** - Getting issue details... STATUS

10) Enforce CI checks for pull requests?

- robert made a proposal some time ago to classify all sling modules in 4 categories (critical, core, contrib, experimental) and apply different levels of enforcements on the PR process
- This was discussed and thought to be a bit too much. Also, e.g. Sonar checks are currently not flawless enough to make them pass mandatory.
- But it was agreed that having enforced reviews in the PR process for the critical modules e.g. Engine, ResourceResolver is a good thing. all changes there can break a lot of stuff.
- To be checked which additional modules should be go into that "critical" list
- Directly pushing to master is not prohibited, but should never be done for the critical modules (except as part of the release process)
- Probably we should update our documentation for new committers a bit also to give them convenience what process to use for PRs in non-critical modules (e.g. create a PR, wait at least 5 days, and if there is no objection that merge it)

12) Is Sling becoming too fat?

- the XSS bundle is "ugly" and definitly fat (although for reasons). Its used a lot in downstream products and also customer projects, so it's unlikely we can get rid of it. But we should try to reduce the usage of it inside Sling as much as possible, to make it optional for "minimal Sling" distributions.
  Carsten Ziegeler will have a look at the remaining usages, for caconfig Stefan Seifert can help
- Joerg Hoh will also check over overall list of current sling modules if there are some more which should be marked as archived but are not marked as that currently

Key	Summary	T	Created	Updated	Due	Assignee	Reporter	P	Status	Resolution

SLING-12060	Remove dependency to Sling osgi.commons	⬆	Oct 01, 2023	Nov 13, 2023	Carsten Ziegeler	Carsten Ziegeler	⏫	CLOSED	Fixed
SLING-12059	Make dependency to metrics optional	⬆	Sep 30, 2023	Oct 09, 2023	Carsten Ziegeler	Carsten Ziegeler	⏫	CLOSED	Fixed
SLING-12055	Remove dependency on Sling XSS	⬆	Sep 29, 2023	Oct 08, 2023	Carsten Ziegeler	Carsten Ziegeler	⏫	CLOSED	Fixed

3 issues

13) Cleanup Sling Starter

- we should review the current features for the Starter if we can remove some entries which are now obsolete (e.g. "saaj")
- some of them where introduced for backward compatibility with older Java version and can be dropped probably

Historical references: https://github.com/apache/sling-org-apache-sling-starter/commit/873905db6c0241c4dcc8b016d7b7e2bf60ff7270 / **SLING-7235** - Getting issue details... STATUS

Karl Pauls

14) Create a shared/reusable Sets of Sonar Rules

- Currently, we are using the default Java rules from Sonar, which is not always helpful as they are not tuned for the "OSGi world". Olli did some work on individual modules to define exceptions for individual modules (e.g. using property files)
- It would be nice to create one or two Quality Profiles in sonarcloud.io fitting to most of our use cases in sling which we can fine-tune over time (e.g. one profile for OSGi bundles, and one for non-OSGi modules, e.g. the Maven plugins)

Robert Munteanu will get in touch with ASF infra if there is a possibility to auto-assign certain quality profiles to all of our sling modules without doing this manually

- Asked Fabrice Bellingard, our SonarCloud contact, at **SLING-10506** - Getting issue details... STATUS

Stefan Seifert can help setting up the quality profiles based on those used for wcm.io OSGi modules

15) Change convention for Sling-Initial-Content folder

- As described in SLING-8736 we should change the folder name which is used by convention for the Sling-Initial-Content inside the OSGi bundles
- the name "SLING-INF" is confusing, as it contains actual content and not just metadata as it's the case e.g. for OSGI-INF
- the proposal is to rename the folder to "initial-content"

Oliver Lietz

16) Switch to GitHub issues?

- ASF provides two bug trackers (bugzilla and JIRA), but it's also allowed to use GitHub issues directly
- does it make sense for Sling to switch to github issues instead of jira?
- probably easier to find for users, but not so easy to find the right module for them. it's possible to transfer issues https://docs.github.com/en/issues/tracking-your-work-with-issues/transferring-an-issue-to-another-repository in case it was mis-placed.
- what process should be used then for release notes/release tagging? version management more difficult - milestones for versions should work fine?
- there is only one issue type - labels have to be used. there are default labels
- issue which affects multiple modules is not so nice - but that should be doable
- A second step after this switch would be to think about partially automating the release process. The actual creation of the maven release probably has to be kept locally due to the signing requirement, but other steps should be able to automate.

17) Use Sling for our website?

- This was a bit controversial, but some see it beneficial if we try to use Sling for our own website (eat your own dog food) - either as part of a static site generation process, or running actively.
- Pros and cons discussed, technically is should be doable

Karl Pauls