Scaling Sling Development

STATUS: PROPOSAL

Motivation

Apache Sling is a highly modular project, consisting of 290+ modules as of 06 Mar 2019 . As such, there is extra work associated with releases and keeping tabs on community contributions compared to more monolithic projects. Additionally, onboarding new contributors is a process that should be simplified. Removing barriers from contribution is a great way to allow contributors to focus on getting actual contributions done, rather than finding out how to contribute.

This document lists a number of perceived pain points for Sling contributors, developers and release managers and proposes solutions to improve the situation.

PRB-CONTRIB: slow turnaround for pull requests

Pull requests are often slowly validated and merged, mainly due to lack of reviewer time. Besides reviewing the code there is also a need of running the build, integrating in the Sling Starter and running the integration tests. This often becomes time consuming for reviewers and causes delays for the contributors.

Currently pull requests are validated by:

- running the Maven build
- running an incremental SonarQube analysis

Enhanced pull request validation *for bundles included in the Sling starter* would include:

- deployment of current SNAPSHOT in the latest Sling Starter
- execution of Sling ITs against the latest Sling Starter + module SNAPSHOT from pull request

Additionally, pull request validation should be enhanced to also report code coverage changes.

This would increase confidence of reviewers and reduce turnaround time for pull requests.

David Bosschaert rightly points out that

> *it can often happen that people who could apply pull requests simply missed them because they were busy looking elsewhere. One thing that sometimes helps in such cases is to simply ping someone directly involved in the specific area and ask them if they have time to merge the PR. I tend to think that the personal approach (rather than an automated email) sometimes has more effect.*

To spot those stale PRs we can consider building a dashboard of open pull requests. Alternatively, an automated email once per week to dev@sling mightalso work (or might get ignored).

Stefan Seifert added that if we decide to send emails we should

> *make sure that those PRs which were added since the last mail report are listed separately from the old ones to easily spot what is new.*

PRB-RELEASE: manual work for Sling releases

The Sling release management page lists a large number of manual steps that are required to perform a release. Many of these must be run by the release manager, since by definition releases are individual acts in the Apache Software Foundation. However, the steps can be automated or simplified by tooling.

Step	Tools	Automation potential	Status
Staging release candidates	Maven, GPG	Small, iusually a single Maven command. Requires access to developer's Maven credentials and GPG keys	Not planned
Starting the vote	Email, Jira	Medium, boilerplate text where placeholders are replaced	✓ **SLING-8392** - Create sub-command to manage the Jira update when promoting a release OPEN **SLING-8311** - Investigate creating a Sling CLI tool for development task automation RESOLVED

Counting the votes	Email	Medium, reading emails and composing a release result mail	✓ SLING-8311 - Investigate creating a Sling CLI tool for development task automation RESOLVED
Promotion - dist. apache.org	SVN	Small, adding/removing files from SVN.	?
Promotion - register with Apache Reporter Service	HTTP	Large, script already exists	✓ SLING-8311 - Investigate creating a Sling CLI tool for development task automation RESOLVED
Promotion - push to Maven Central	HTTP	Medium, interaction with Nexus	SLING-8338 - Create sub-command to manage the Nexus stage repository release when promoting a release OPEN
Promotion - update site	Git	Large - modifications are mostly automatic, a pull request can be automatically issued	Planned SLING-8393 - Create sub-command to update the Sling website when a release is made OPEN
Jira Update	Jira	Large - releasing current version, releasing new version, closing issues fixed in current version.	SLING-8392 - Create sub-command to manage the Jira update when promoting a release OPEN
Sling Starter update	Git	Large - a pull request can be created when outdated Sling bundles are found in the Launchpad	Planned SLING-8394 - Create sub-command to update the Sling starter when a release is made OPEN

The proposed solution comes with two parts:

1. A Sling-Dev CLI tool that automates tasks that must be manually performed by the developer: Email generation, Jira management, etc.
2. A Sling Bot that will automatically issue pull requests for the website and starter repositories.

David Bosschaert has suggested that

> *It would also be interesting to see if we can get an automated process to move released binaries to dist so that non-PMC members who have done a release don't depend on PMC members for that step.*

An investigation for a tool has started at SLING-8311 - Investigate creating a Sling CLI tool for development task automation RESOLVED

PRB-ONBOARD: lack of developer onboarding for Sling developers

When contributors approach Sling, especially from an AEM background, there is a lack of transparency regarding what Sling development practices are:

- code conventions
- automated tests
- local testing (Sling starter vs Quickstart, Composum vs CRX DE Lite,initial content in bundles vs content packages)

Note that while content packages are usable with the Sling Starter, we store the initial content in bundles.

These should be better explained as part of an 'onboarding' contributors page

PRB-GHDOC: documentation on the website is disconnected from the one on Github

There are currently no links from the GitHub projects to the website and sometimes the documentation is maintained in one place and sometimes in another. This is not consistent and lacks usability.

We should have ways of

- linking from the GitHub repository to the site documentation (if exists)
- linking from the site documentation to the GitHub repository

Additionally, it would be interesting to find out if we can reuse the same documentation for modules there it makes sense, i.e. self-contained ones.