Using Git with Sling

This page documents decisions we made while finding out how to best work with Git as well and various quality-of-life tips related to Git.

Cloning the Sling git repositories

This is documented in the Sling Aggregator Github module.

Accepting new contributions

When accepting an external contribution we will import it into the `sling-whiteboard` repository. Before making a new release, we will move it to its own repository with the release manager also proposing a proper artifact id and therefore repository name.

It is recommended to provide ownership information when accepting new contributions by maintaining the original *Author* git header.

See also dev@sling - Where do we put new git modules?

Code donations

In case of code developed externally and not initially intended to be hosted at the ASF IP clearance is required. The general rules are outlined at http://incubator.apache.org/ip-clearance/ . For a hands-on example, the Sling Journal-based Content Distribution is a good example:

- Incubator IP clearance page: http://incubator.apache.org/ip-clearance/sling-distribution-journal.html
- Top-level Jira: **SLING-8326** - Donation proposal of Journal based Sling Content Distribution CLOSED
  - IP clearance sub-task: **SLING-8345** - IP clearance of Journal based Sling Content Distribution donation RESOLVED
  - Source code import sub-task: **SLING-8346** - Import Journal based Sling Content Distribution source code RESOLVED

Creating new repositories

Git repository creation

Repository creation is open only to PMC members. If you are not part of the PMC please ask on dev@sling.apache.org and someone will do this for you.

When creating new repositories we follow the convention of deriving the repository name from the Maven artifactId. The full logic can be found in the migrate-to-git.sh script, but the gist is:

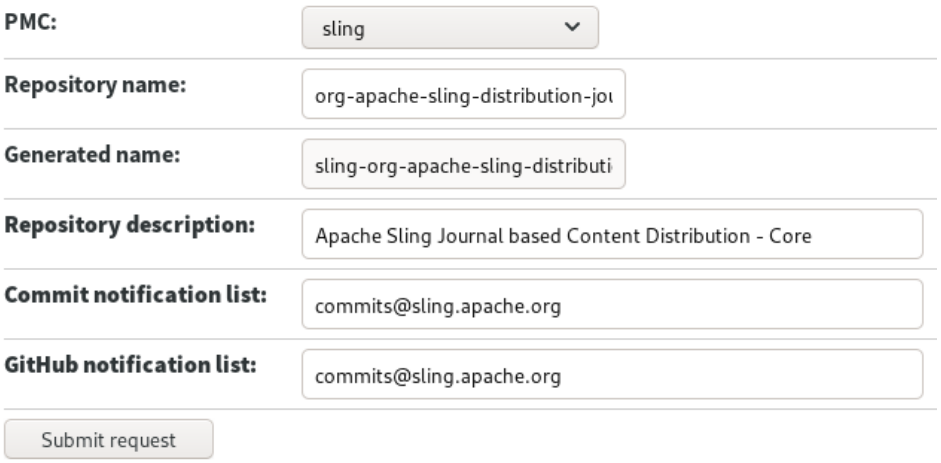
1. Append `sling-` to the artifact id if not there (required by ASF infra conventions)
2. Replace all dots with dashes (ASF infra does not allow dots in repository names)

It is recommended to announce the intention on the mailing list, without calling a formal vote, since renaming repositories requires ASF infra manual intervention.

Creating a new repository is done using the web UI at https://gitbox.apache.org/ . Please fill in the fields in the following manner:

- *Name*: artifact Id , dots replaced with slashes. Should not contain the `sling-` prefix as the form will auto-insert it in the 'Generated Name' field
- *Repository Description*: `name` from the pom.xml, otherwise short summary of the form *Apache Sling Foo*
- *Commit Notification List:* commits@sling.apache.org
- *GitHub Notification List*: commits@sling.apache.org

See a recent example below



Permission propagation

The scripts will be created in gitbox and github at the same time, but permissions will be granted to the GitHub repository sometime later, due to the way ASF infra tooling works. If permissions don't appear in two hours, raise an INFRA ticket. In the meantime you can use the gitbox backend to push your changes.

Importing existing code from the whiteboard

If you have code that exists in the whiteboard and you want to move it to a separate repository it is recommended to keep the history. The steps to achieve that are below, given that `${FOLDER}` is the folder from the Sling Whiteboard that you want to move to a new repository, and that `${NEW_REPO}` is the name of the new repository. The new repository should already be created.

1. Clone the whiteboard repository

```
$ git clone https://github.com/apache/sling-whiteboard.git
$ cd sling-whiteboard
```

2. Reduce the repository to only the subfolder you're interested in

```
$ git filter-branch --prune-empty --subdirectory-filter ${FOLDER}
```

3. Change the origin remote URL to the desired one

```
$ git remote set-url origin https://github.com/apache/sling-${NEW_REPO}.git
```

4. Review your changes and push them

```
$ ls -1
$ git log
$ git push -u origin master
```

Reference: https://help.github.com/en/articles/splitting-a-subfolder-out-into-a-new-repository

Update the repo manifest

Regenerate the default.xml manifest form the sling-aggregator repo using `groovy collect-sling-repos.groovy > default.xml`.

Boilerplate files

Ensure that at a minimum the repository contains the following files:

- README.md - brief description of the module
- LICENSE - the standard Apache-2.0 license file (available in the sling-aggregator repo)
- CODE_OF_CONDUCT.md - reference to the Apache Software Foundation code of conduct (available in the sling-aggregator repo)
- CONTRIBUTING.md - reference to the Apache Sling contribution guidelines (available in the sling-aggregator repo)

- Jenkinsfile - delegates to the Apache Sling Pipeline library (example available in all repos, for instance see the Jenkinsfile for the sling-org-apache-sling-api repository)
- .gitignore

Enroll in Kibble

Add the repository as a source in the Apache Kibble demo instance to make sure development statistics are collected from it. Note that the demo user does not have rights to add new source, so either sign up with a new user or ask on dev@sling.apache.org for someone to add the repositories.

Onboard to SonarCloud

See the onboarding section in SonarCloud analysis .

Creating links to Git commits

To quickly create links to Git commits in JIRA a script such as the one below can be used:

jira_link_for_commit.sh

```
#!/bin/sh -e
if [ $# -eq 0 ]; then
    commit=HEAD
else
    commit=$1
fi
hash=$(git rev-parse --short ${commit})
base=$(git remote get-url origin)
url=${base%.git}/commit/${hash}
echo "[commit ${hash}|${url}]"
```