Migrating the Sling Starter to the Feature Model

STATUS: IMPLEMENTED

There have been many discussion about moving the Sling Starter to the Feature Model. This page attempts to summarise the current state of affairs and propose a plan for switching over.

Table of contents

Motivation

The Feature Model has seen a lot of activity and improvements during the last two years and is actively developed. It better tooling support, for instance a significant number of feature model analysers that provide fast feedback on the application model without starting it. The feature model launcher supports faster turnarounds as it is able to launch based of the configuration files alone, without the need to create an executable jar and extract its contents.

The launcher is pure OSGi launcher and the features being discussed in the OSGi expert groups as part of RFP 188 and RFC 241 .

Known gaps

G1: There is no support for WAR deployments. It remains to be seen whether if and when we should look into this.
G2: There are no plans to support a 'fat' jar which embeds all of the required bundles, configurations, etc. Instead, we want to provide a launcher and a reference to feature model definition (archives or plain model files).

Impact

Switching to the feature model does not only affect the Sling Starter module itself. The following components are also affected:

- Launchpad Testing . Sling will have to be launched with the feature model launcher for integration tests
- Docker Image. The Docker image will have to be adjusted to use the feature launcher
- Maven Archetypes
- Documentation
- Other testing projects consuming the Sling Starter

There should be no impact on OSGi bundles and content packages.

Rollout Steps

Phase 1

This phase will keep the provisioning model as the main output and produce feature model files as a secondary aggregate, for early integration and testing.

Steps:

1. Announce intention to switch and drop WAR support on user@sling (also Twitter and other channels). This will allow us to understand how important WAR support is for our consumers and maybe find parties interested in contributing and maintaining it. (**SLING-9497** - Getting issue details... STATUS)
2. Starter: Create feature model aggregates on the fly from the build - oak_tar and oak_mongo. This should use the feature-converter-maven-plugin to create the feature files. (**SLING-9498** - Getting issue details... STATUS)
3. Launchpad Testing: create branch running ITs against the oak_tar aggregate created in the previous step (**SLING-9499** - Getting issue details... STATUS)

Success criteria:

1. Conversation on user@sling.apache.org settled down with clear outcome regarding war support
2. Sling Starter creates and deploys oak_tar and oak_mongo feature model aggregates
3. Launchpad Testing branch against feature model passes all tests

Phase 2

This phase will switch the starter and the launchpad-testing module to the feature model. At this point there should be no more provisioning model files used in these projects.

Steps:

1. Fully switch starter to feature model SLING-9595 - Getting issue details... STATUS .
    a. The feature model converter will be used to create individual feature files out of the provisioning model files
    b. Feature model aggregates will be defined for the current run modes: oak_tar and oak_mongo
    c. Feature model analysers will be configured for the main aggregates
    d. Produce feature archives for oak_tar and oak_mongo
2. Switch launchpad testing to provisioning model SLING-9596 - Getting issue details... STATUS
    a. provisioning files switched to feature files
    b. run against the starter using feature model

Success criteria:

1. Sling Starter: all provisioning model files removed
2. Sling Starter: feature model analysers configured
3. Sling Starter: produce oak_tar and oak_mongo aggregates
4. Sling Starter: produce feature archives
5. Launchpad Testing: feature model testing branch merged into master
6. Launchpad Testing: all tests passing

Phase 3

This phase runs contains all steps following the main feature model conversion. These can run in parallel and should be done to consider the switch to the feature model complete.

1. Refine feature model files. The Phase 2 conversion does not have to be perfect in terms of feature model definitions. In this phase we will refine the feature definitions to ensure that the features are consistent and to get any benefits in terms of reuse for multiple aggregates SLING-9636 - Getting issue details... STATUS
    a. Separate note: we do not use variables to define e.g. shared versions. We should definitely do this.
2. Remove or refine smoke IT for the starter project. Right now it has been disabled since we have the analysers running, but they don't check that the repository actually starts up. SLING-9637 - Getting issue details... STATUS
3. Produce a kickstart jar SLING-9635 - Getting issue details... STATUS
4. Update docker image. The docker image will now use the feature model launcher. SLING-9638 - Getting issue details... STATUS
5. Update archetypes SLING-9641 - Getting issue details... STATUS
    a. Sling Project Archetype. This archetype is a very good candidate for creating a feature-model based application
    b. New Sling-Feature-Archetype . We may decide to also create a standalone archetype for a feature model application, similar to the Slingstart Archetype
6. Retire no longer needed projects SLING-9639 - Getting issue details... STATUS
    a. launchpad-testing-bundles
    b. lauchpad-testing-war
7. Update documentation
    a. Sling Site.
        i. There is extensive documentation on launching Sling on the website. These should be adjusted to the feature model launcher SLING-9640 - Getting issue details... STATUS
        ii. The provisioning model documentation should include a note that the feature model is now the preferred mechanism SLING-9642 - Getting issue details... STATUS
        iii. The feature model documentation should be enhanced. We could either move the documentation from GitHub to the website, or provide a better summary + links to all modules SLING-9643 - Getting issue details... STATUS
    b. What Else?

Success Criteria:

1. Sling Starter: feature model definitions reviewed and refined
2. Sling Starter: produce kickstart jar
3. Docker image: built using feature model launcher
4. Archetypes: Sling Project Archetype updated to use the feature model
5. Archetypes: New Sling-Feature-Archetype created
6. Launchpad Testing: unneeded projects retired
7. Sling Site: Reviewed for documentation to update
8. Sling Site: Documentation on launching Sling is updated to reflect feature model
9. Sling Site: Provisioning Model documentation updated to reflect preference for feature model
10. Sling Site: Feature Model Documentation Enhanced
11. Sling Site: Other remaining pieces of documentation enhanced
12. Sling Site: Tutorials updated to reflect Feature Model launchers

Official way of launching Sling

DRAFT

With the provisioning model and launchpad setups for the Sling Starter we always had a all-in

Out of scope

The following topics are explicitly out of scope, to ensure that we focus on the initial switch

- moving the launchpad-testing module to the Sling starter repository
- moving the Docker image build process to the Sling starter repository

We may decide to pursue these in a later phase, after the migration to the feature model is complete.

Open topics

- Should we create a Docker Image for the feature launcher, for convenience? Could achieve zero in-advance download scenario with remote feature files or feature archives
- What options for reuse/integration do we have with the Sling Karaf Features?
- What do we do for projects using the Sling Starter as a base? (e.g. org-apache-sling-scripting-bundle-tracker-it, org-apache-sling-nosql-launchpad , org-apache-sling-scripting-sightly-testing)
    - We could keep producing a slingstart provisioning-model based jar, or maybe just the model files
    - Alternatively, these projects could migrate, if desired, at their own pace

Resolved topics

- WAR support was dropped, no one replied saying they want it

References

1. dev@sling.apache.org - [hackathon] switch to feature model (2019/09)
2. dev@sling.apache.org - Sling FM Starter Module (2019/11)
3. dev@sling.apache.org - sling12 timeline/feature models (2019/11)
4. dev@sling.apache.org - [RT] Feature archives, launcher and Sling Starter (2020/03)
5. dev@sling.apache.org - [Discuss] (SLING-8350) Switch the Sling starter to the feature model (2020/05)
6. **SLING-8350** - Getting issue details... STATUS (2019/04)