This document provides more in-depth information about the Spring Artifactory. If you're looking for more basic instructions about using Spring Framework jars in your application, see [[Spring Framework Artifacts]].

Overview

The Spring Artifactory is a repository manager created and hosted by JFrog on behalf of Spring. It provides one place to search all Spring open-source project artifacts such as binary, source, and Javadoc jars, distribution zips, and others. At the same time it provides one place for Spring open-source projects to get the external dependencies they need to build with.

The Artifactory provides a Web UI and a REST API. In addition project team members can place watches on individual artifacts, directories, or entire repositories, which can be useful to track project releases.

Local Repositories

Local repositories host artifacts produced directly by Spring projects: * libs-snapshot-local - snapshot artifacts versions * libs-milestone-local - milestones and release candidate versions * libs-release-local - release versions

Remote Repositories

Remote repositories serve as caches for artifacts hosted elsewhere such as Maven Central, JBoss, or any others. These repositories are not typically accessed directly, but rather are *aggregated* by virtual repositories as discussed below.

Virtual Repositories

Virtual repositories aggregate any combination of local and remote repositories to provide convenient and controlled access to artifacts. If you want to see what repositories a virtual repository is composed of, to to https://repo.spring.io and click on the link to Artifacts (on the left).

The following are virtual repositories for use in Spring applications: * snapshot - snapshot artifacts versions * milestone - milestones and release candidate versions * release - release versions

The following are virtual repositories for use by Spring open-source projects to resolve Spring project artifacts and also their downstream dependencies:

- libs-release – release versions
- libs-milestone - release, RC, and milestone versions
- libs-snapshot - release, RC, milestone, and snapshot versions

Note that this is designed in such a way that build scripts need only point to a single repository (vs adding multiple). For example by switching from `libs-release` to `libs-milestone`, you can use not only all release versions,

but also milestones and release candidates. Or by pointing to `libs-snapshot` you get access to the full set of Spring artifacts and downstream dependencies. This results in fewer HTTP requests and allows the server do the work.

Spring applications can also use the `libs-*` repositories but note that for security reasons, only authenticated users (i.e. Spring project committers), may cause a remote repository to resolve and cache new dependencies. That means non-authenticated users may be able to resolve *every* dependency the need against repo.spring.io, and will fall back to Maven Central or other third-party repositories.

There also exist plugins-snapshot-local and plugins-release-local for custom plugins created by the Spring team.

Adding a Repository

If necessary a project lead or committer can request to have a new repository added. Please verify the artifact does not exist by performing various searches on https://repo.spring.io.

Occasionally a dependency is not unavailable in any Maven repository and must be downloaded manually or built from source. In such cases, you may upload these artifacts directly into the `ext-snapshot-local` or `ext-release-local` repositories using the "Deploy" tab available from the homepage at https://repo.spring.io. The wizard there will walk you through POM generation, specifying coordinates, etc.

IMPORTANT: Please understand the license of any artifacts you upload in this fashion. If the license does not permit distribution, you cannot use these repositories. `ext-private-local` *exists for this purpose, and makes these artifacts available to internal users only.*