Releasing IG

This page collect and organize operations to be performed when releasing IG.

Step-by-step guide

Branching

1. Kindly ask the Release Engineering Team for permissions to directly push on the main repository
2. Execute maven release plugin to handle branching:
    a. When branching of the `master`, without touching the `master`'s version

```
mvn release:branch \
    -Pproduction-backstage-connect-build \
    -DbranchName=6.1 \
    -DupdateBranchVersions=true \
    -DupdateWorkingCopyVersions=false \
    -DautoVersionSubmodules=true \
    -DreleaseVersion=6.1.0-SNAPSHOT
```

    b. When branching the master, updating version on the `master` branch only

```
mvn release:branch \
    -Pproduction-backstage-connect-build \
    -DbranchName=6.5 \
    -DupdateWorkingCopyVersions=true \
    -DautoVersionSubmodules=true \
    -DdevelopmentVersion=7.0.0-SNAPSHOT
```

Updating dependencies to expected version

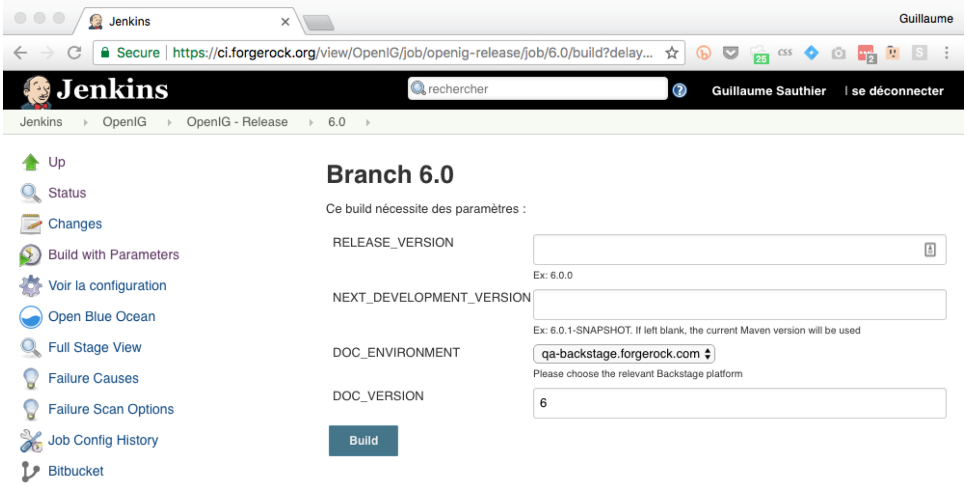
1. Verify IG is using the appropriate `openig-toolkit`
2. Verify that IG is using the appropriate `forgerock-bom` (COMMONS)
3. Update, commit and merge
4. Check versions in `jenkins-scripts/modules/functional-tests-linux.groovy`

Manual copyright checking

1. Such as footers in HTML pages (sample app, welcome page, studio)

Release

1. Open IG Release's Pipeline (https://ci.forgerock.org/view/OpenIG/job/openig-release/)
2. Select the branch you want to release
    a. If it doesn't appear, execute "`Scan Multibranch Pipeline Now`", the release branch should show up some time later
3. When on the branch page, execute "`Build with Parameters`"



Branch 6.0

Ce build nécessite des paramètres :

RELEASE_VERSION

Ex: 6.0.0

NEXT_DEVELOPMENT_VERSION

Ex: 6.0.1-SNAPSHOT. If left blank, the current Maven version will be used

DOC_ENVIRONMENT: qa-backstage.forgerock.com

Please choose the relevant Backstage platform

DOC_VERSION: 6

When about to release version x.y.z :

    a. Release Version: Name of the tag and released version, should be of the form x.y.z or x.y.z-RCi (example : 6.1.0-RC2)
    b. Next Dev Version: Next version (usually, the same version that you are about to release) ending with -SNAPSHOT (ex: 6.1.0-SNAPSHOT)
    c. Doc Env: Choose QA for intermediate builds (milestones and RCs), that's the default, choose backstage for the real release only
    d. Doc Version: short version name ie x or x.y (example : 6 for 6.0.0 and 6.1 for 6.1.0)

Documentation

The IG documentation is separated from the project codebase (different lifecycle), but has a dependency on it (extract some files from the project archive).

1. Verify that `softwareVersion` property is set to the associated IG maven-release version
2. Verify doc version properties (pay attention, some of them are also declared in `forgerock-release` profile)

   ```
   <softwareVersion>6.5.0-SNAPSHOT</softwareVersion>
   <softwareFullDisplayVersion>6.5.0</softwareFullDisplayVersion>
   <docReleaseVersion>6.5</docReleaseVersion>
   <platform.vers>6.0.0-SNAPSHOT</platform.vers>
   <ig.vers>${platform.vers}</ig.vers>
   <am.vers>${platform.vers}</am.vers>
   <ds.vers>${platform.vers}</ds.vers>
   <jpa.vers>3.5</jpa.vers>
   <amster.vers>${platform.vers}</amster.vers>
   ```

3. Go to the Jenkins doc pipeline (https://ci.forgerock.org/view/Docs/job/docs-pipelines/)
4. Depending if you want to do an intermediate release or a final one, select either product-docs-qa-release or product-docs-release (respectively)
   a. As in IG release pipeline, execute Build with Parameters
      i. Select the appropriate Doc Set (product name + expected version)
         1. If the doc-set does not exist yet, ask Mark Craig
      ii. Use the appropriate doc version number (6.1.0 or 6.1.0-1 for subsequent fix releases)

Related articles

- Create Sprint ID Cloud Branches
- Releasing IG