# New release guidelines

This document provides:

- This document contains a mixture of information, advice and examples of best practices for releasing versions of Apache Juneau.

The instructions provided here are consistent with, but not a replacement for the **ASF Release Guidelines**.



## 1 - Intro

The Apache Juneau project uses the Nexus installation at the Apache Software Foundation to stage Maven artifacts before the release and later publish them; the POM in the root project contains all the necessary configuration.  This guide covers deploying to Nexus using Maven.  For alternatives see the **ASF wide documentation**.  If this is the first time you are publishing to the ASF's Nexus instance you'll need to **prepare your development environment**.

Juneau components are expected to use Maven to build the jar/zip files.  The version of Maven used is assumed to be Maven 3 throughout.  At a minimum, Juneau releases **must** include full source distributions packaged in zip archives.

This document assumes that the release is being prepared on a linux/unix system, and that steps are being executed from the command-line.  The principles are the same, however, for a release prepared on other operating systems.

These instructions assume you are taking on the role of release manager.  This implies you are at least one of the developers.

If you're using macOS, you'll want to install **wget**.

Yellow blocks are code blocks that can be executed as-is UNLESS you see red text indicating values that need to be replaced.

## 2 - Preparation

### 2.1 - Determine version number

Consult the **versioning guidelines** and check that the current level of compatibility is suitable for the proposed version number.

### 2.2 - Propose email to dev@juneau

Send an email to the **dev@juneau** mailing list informing the community you plan on creating a new release.  The message can be simple:

**To: dev@juneau.apache.org**

**[PROPOSE] Release Apache Juneau 9.x.x RCx**

Hi everyone,

I'm going to create a new release of Juneau because <state short reason why>.

If I do not hear any objections within the next 24 hours, I'm going to go ahead and start the process.

The release will be called 9.x.x-RCx. The next release will be set to 7.x.y.

# 3 - Creating a release candidate

## 3.1 - Create new release in JIRA

Log into **JIRA** and create the new version (e.g. **7.0.1**). Change the **Release date** on the current version to today.

Generate the release notes for the current release as plain text. Click on the current release and click the **Release Notes** button. Click the **Configure Release Notes** button to select plain text. Copy the contents of this file to the root **RELEASE-NOTES.txt** file and commit the change.

## 3.2 - Update Release Notes section in overview.html

We maintain externally-consumable new-and-noteworthy information for all our releases in our javadocs.

Using the release notes above as a guide, create release notes in the **Release Notes** section of the **/juneau-doc/src/main/javadoc/overview.html** document. Use the previous release as a guideline.

## 3.3 - Update LICENSE and NOTICE files

Check the contents of the **LICENSE** and **NOTICE** files in the root directory.

Ensure that the LICENSE files contain the latest information specified in the **Apache Licenses** page.

Check that the years in the copyright statement in the NOTICE file are correct. The notice file should contain the following:

```
Apache Juneau
Copyright 2016-{this-year} The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).
```

Developer documentation on how to apply the Apache License can be found in **Applying the Apache License, Version 2.0** and **ASF Source Header and Copyright Notice Policy.**

## 3.4 - Run release script

If you've already successfully performed a release, you can run the automated shell script to perform the release. If you do not wish to use the script, skip to section **3.5 - Prepare your build machine**.

The release script consists of the following files:

- **juneau-release-env.sh**
- **juneau-release.sh**

Update the contents of the **juneau-release-env.sh** to match the values for the new release:

```
export X_VERSION=7.0.0
export X_NEXT_VERSION=7.0.1-SNAPSHOT
export X_RELEASE=juneau-7.0.0-RC1
export X_STAGING=~/tmp/dist-release-juneau
export X_USERNAME=<your apache username>
export X_EMAIL=you@apache.org
```

Make sure not to check in this changed file!

Next, run the **juneau-release.sh** file from the juneau root directory. It will walk you through the entire process.

If successful, you can skip to section **4 - Voting.**

## 3.5 - Prepare your build machine

Run the following commands to set environment variables that will be used throughout this page.
Set **X_RELEASE** to the version you plan on release and **X_STAGING** to the staging directory to use on your machine:

```
export X_VERSION=7.0.0
export X_NEXT_VERSION=7.0.1-SNAPSHOT
export X_RELEASE=juneau-7.0.0-RC1
export X_STAGING=~/tmp/dist-release-juneau
export X_USERNAME=<your apache username>
export X_EMAIL=you@apache.org
```

Run the following commands:

```
cd ~/.m2
mv repository repository-old
rm -rf repository-old &
rm -rf $X_STAGING
mkdir -p $X_STAGING
mkdir $X_STAGING/git
cd $X_STAGING/git
git clone https://gitbox.apache.org/repos/asf/juneau.git
git clone https://gitbox.apache.org/repos/asf/juneau-website.git
cd juneau
git config user.name $X_USERNAME
git config user.email $X_EMAIL
```

Make sure you're running at least Java 17 and Maven 3:

```
java -version
mvn -version
```

If you're not running Java 8, you'll get a "javadoc: error - invalid flag: -Xdoclint:none" when trying to generate the Javadocs.

## 3.6 - Update KEYS file if necessary

Check that the **KEYS** file contains your correct key and fingerprint.  If not, then follow the instructions in this section.

You need to have **gpg** and preferably a **GPG Agent** installed on the machine you will build the release on. This needs to be configured with your **GPG release key**.

Find your PGP fingerprint:

```
gpg --fingerprint $X_EMAIL
```

You should see something like the following...

```
jamesbognar-ltm:dist-release-juneau james.bognar$ gpg --fingerprint jamesbognar@apache.org
pub   rsa2048 2016-09-21 [SC]
      59E1 A375 4EF6 0E6F CE42  ABFE 3629 2BD2 BA7D 3A86
uid           [ultimate] James Bognar <jamesbognar@apache.org>
sub   rsa2048 2016-09-21 [E]
```

Your key should list your **@apache.org email address**. Also check your key has **not expired** - you can use **gpg --edit-key** and **gpg --send-key** to update.

The public key must first be **uploaded to a public keyserver**.  To do this, take the *last* 8 digits from the hex **fingerprint** and upload them with the following command:

```
gpg --keyserver pgp.mit.edu --send-key BA7D3A86
```

Edit your details on **id.apache.org** to provide your **OpenPGP Public Key Primary Fingerprint**, e.g.:

```
59E1 A375 4EF6 0E6F CE42  ABFE 3629 2BD2 BA7D 3A86
```

The following is outdated:

> (people.apache.org will fetch your public key from the key servers - you may need to allow some hours if you are adding a new key)
>
> Verify that the **people.apache.org juneau.asc** file now includes the correct key fingerprint (scroll through or test with gpg --import), and update on dist:
>
> ```
> cd $X_STAGING
> svn co https://dist.apache.org/repos/dist/release/juneau/
> cd juneau
> rm KEYS
> curl https://people.apache.org/keys/group/juneau.asc -o KEYS
> svn commit -m "Updated KEYS" KEYS
> ```

Instead, you can use the Signing Releases guide and run:

```
gpg --list-sigs "$X_EMAIL" >> KEYS && gpg -armor --export "$X_EMAIL" >> KEYS
```

## 3.7 - Verify build

Now we'll make sure they build normally, *pass all the tests* and complete with **BUILD SUCCESS**:

```
cd $X_STAGING/git/juneau
mvn clean verify
```

From the project root, generate the javadocs and make sure there are no errors or warnings:

```
mvn javadoc:aggregate
```

Find the following zip files and make sure they deploy correctly in a fresh Eclipse workspace:

- **juneau/juneau-microservice/juneau-microservice-template/target/my-microservice-<version>.bin.zip**
- **juneau/juneau-examples/juneau-examples-rest/target/juneau-examples-rest-<version>.bin.zip**
- **juneau/juneau-examples/juneau-examples-rest-springboot/target/juneau-examples-rest-springboot-<version>.bin.zip**

## 3.8 - Maven repository access

When releasing a stable version, Maven will deploy to Apache's **Maven repository**, a Nexus instance. Here, a **staging repository** will be automatically created.

Ensure you can log in to the Nexus instance before performing a release.

To provide the Nexus credentials, edit your **~/.m2/settings.xml** to include your **apache.org** committer credentials for the servers **apache.snapshots.https** and **apache.releases.https**:

```
<?xml version="1.0" encoding="UTF-8"?>
<settings xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.1.0 http://maven.
apache.org/xsd/settings-1.1.0.xsd" xmlns="http://maven.apache.org/SETTINGS/1.1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <servers>
    <server>
      <id>apache.snapshots.https</id>
      <username>johndoe</username>
      <password>fishsoup</password>
    </server>
    <server>
      <id>apache.releases.https</id>
      <username>johndoe</username>
      <password>fishsoup</password>
    </server>
  </servers>
</settings>
```

To check you have the correct credentials set up for write-access to **Apache's Maven repository**, you can deploy the current SNAPSHOT:

```
cd $X_STAGING/git/juneau
mvn deploy
```

## 3.9 - Prepare release candidate

The contents of the binary distribution is configured in the assembly descriptor **juneau-distrib/src/assembly/bin.xml**. Make sure that all (and only) files that should be included in the binary distribution are included in the fileset elements of the descriptor. We use the **Maven release plugin** to release candidates as it ensures a consistent release process.

Run the following command:

```
mvn release:prepare -DautoVersionSubmodules=true -DreleaseVersion=$X_VERSION -Dtag=$X_RELEASE -
DdevelopmentVersion=$X_NEXT_VERSION
```

**Note:** Maven will use **gpg** multiple times to tag and sign the artifacts - so you might want to install and configure a GPG Agent to avoid repeatedly providing your GPG passphrase.

If anything goes wrong at this state, you will need to undo, edit, and commit required changes and start again.

```
mvn release:rollback git commit git push
git tag -d $X_RELEASE
git push origin :refs/tags/$X_RELEASE
```

Use **git diff** against the previous release tag for a rough check for changes:

```
git diff $X_RELEASE
```

### 3.10 - Deploying release candidate

If the preparation was successful, then you should now be able to do:

```
mvn release:perform
```

This will check out the tag, build it, then sign and deploy the packaged source code and compiled JARs to **http://repository.apache.org**.

### 3.11 - Locate the staging repository

On Apache's Nexus instance, locate the **staging repository** for the code you just released.  It should be called something like **orgapachejuneau-1000**.

Check the ***Updated*** time stamp and click to verify its ***Content***.

**Important** - When all artifacts to be deployed are in the staging repository, tick the box next to it and click ***Close***.

**DO NOT CLICK RELEASE YET** - the release candidate must pass ***[VOTE]*** emails on **dev@juneau** before we release.

Once closing has finished (check with *Refresh*), browse to the *URL* of the **staging repository** which should be something like **https://repository.apache.org/content/repositories/orgapachejuneau-1000**.

Set the following environment variable to match the staging repository:

```
export X_REPO=orgapachejuneau-1000
```

### 3.12 - Uploading to dist.apache.org

The release candidate source code, checksums and signatures should be uploaded to the **dev** area of **dist.apache.org** using **svn** - which we'll check out to a new folder.

```
cd $X_STAGING
rm -rf dist
svn co https://dist.apache.org/repos/dist/dev/juneau dist
svn rm dist/source/*
svn rm dist/binaries/*
mkdir dist/source/$X_RELEASE
mkdir dist/binaries/$X_RELEASE
cd $X_STAGING/dist/source/$X_RELEASE
wget -e robots=off --recursive --no-parent --no-directories -A "*-source-release*" https://repository.apache.org
/content/repositories/$X_REPO/org/apache/juneau/
mv juneau-${X_VERSION}-source-release.zip apache-juneau-${X_VERSION}-src.zip
mv juneau-${X_VERSION}-source-release.zip.asc apache-juneau-${X_VERSION}-src.zip.asc
mv juneau-${X_VERSION}-source-release.zip.md5 apache-juneau-${X_VERSION}-src.zip.md5
gpg --print-md SHA512 apache-juneau-${X_VERSION}-src.zip > apache-juneau-${X_VERSION}-src.zip.sha512
rm *.sha1
cd $X_STAGING/dist/binaries/$X_RELEASE
wget -e robots=off --recursive --no-parent --no-directories -A "juneau-distrib*-bin.zip*" https://repository.
apache.org/content/repositories/$X_REPO/org/apache/juneau/
mv juneau-distrib-${X_VERSION}-bin.zip apache-juneau-${X_VERSION}-bin.zip
mv juneau-distrib-${X_VERSION}-bin.zip.asc apache-juneau-${X_VERSION}-bin.zip.asc
mv juneau-distrib-${X_VERSION}-bin.zip.md5 apache-juneau-${X_VERSION}-bin.zip.md5
gpg --print-md SHA512 apache-juneau-${X_VERSION}-bin.zip > apache-juneau-${X_VERSION}-bin.zip.sha512
rm *.sha1
cd $X_STAGING/dist
svn add source/$X_RELEASE
svn add binaries/$X_RELEASE
svn commit -m "$X_RELEASE"
```

Now verify that the files are available on **https://dist.apache.org/repos/dist/dev/juneau** - you might need to refresh your browser.

## 4 - Voting

Any Apache release must be approved through a **vote** of the project's PMC.

The **[VOTE]** threads should be open for at least 72 hours each, allowing time for sufficient review and catering for differences in holidays and time zones.

Including 24 hours grace period for the download mirrors to update. (For urgent security fixes it might be advisable to do concurrent votes).

Anyone in the **community** can participate in the review and vote, not just PMC members or committers.  Only votes from PMC members are binding.

### 4.1 - Vote email to dev@juneau

To start the release vote, modify the below example and send to **dev@juneau**.  Remember **[VOTE]** in the subject line and change the version number and the checksums to match the release candidate.

**To: dev@juneau.apache.org**

**[VOTE] Release Apache Juneau 9.x.x RCx**

I am pleased to be calling this vote for the source release of Apache Juneau 9.x.x RCx.

The binaries are available at:

https://dist.apache.org/repos/dist/dev/juneau/binaries/juneau-9.x.x-RCx/

The release candidate to be voted over is available at:
https://dist.apache.org/repos/dist/dev/juneau/source/juneau-9.x.x-RCx/

SHA-512 checksums:
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa apache-juneau-9.x.x-src.zip

aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa apache-juneau-9.x.x-bin.zip

Build the release candidate using:

mvn clean install

The release candidate is signed with a GPG key available at:
https://dist.apache.org/repos/dist/release/juneau/KEYS

A staged Maven repository is available for review at:
https://repository.apache.org/content/repositories/orgapachejuneau-10xx/

The Git commit for this release is...
https://gitbox.apache.org/repos/asf?p=juneau.git;a=commit;h=dd034037676f28ee89e9ef4b1cabd2bce6e40d34

Please vote on releasing this package as:
Apache Juneau 9.x.x

This vote will be open until 04-May-2016 1:30pm and passes if a majority of at least three +1 Apache Juneau PMC votes are cast.
*(needs to be at least 72 weekday hours)*

[ ] +1 Release this package
[ ] 0 I don't feel strongly about it, but don't object
[ ] -1 Do not release this package because...

Anyone can participate in testing and voting, not just committers, please feel free to try out the release candidate and provide your votes.

## 4.2 - Tallying the vote

After at least 72 hours, count the votes in the **[VOTE]** thread.

To close the vote thread, send a **[RESULT][VOTE]** email to **dev@juneau**, e.g.

**To: dev@juneau.apache.org**

**[RESULT][VOTE] Release Apache Juneau 9.x.x RCx**

Voting for Apache Juneau 9.x.x is now closed. The release has passed with the following tally:

Person 1 (PMC)
Person 2 (PMC)
Person 3

List PMC votes first, then other votes.

If the vote has not passed (less than +3, or more negative than positive) then you will need to pull the release candidate and start over after addressing the concerns.

If there are any -1 or 0 votes, then those concerns should be addressed.  Often it can be sufficient to just track these in Jira as scheduled for the next release - while other times the concern could be grave enough to pull the release (e.g. doesn't compile). Given sufficient votes, as a Release Manager you get to decide if the concerns (including your own) are blocking or not.

# 5 - Validation

Download and extract the **apache-juneau-x.x.x-src.zip** and **apache-juneau-x.x.x-src.zip** files from https://dist.apache.org/repos/dist/dev/juneau/

Make sure the source zip contains the following:

- **LICENSE**
- **NOTICE**
- **RELEASE-NOTES.txt**

The **bin** file should contain the following artifacts:

- **projects**
  - **my-springboot-microservice-x.x.x.zip**
  - **my-jetty-microservice-x.x.x.zip**
  - **juneau-examples-rest-springboot-x.x.x.zip**
  - **juneau-examples-rest-jetty-x.x.x.zip**
  - **juneau-examples-core-x.x.x.zip**
- **osgi**
  - **org.apache.juneau.svl_x.x.x.jar**
  - **org.apache.juneau.rest.server.springboot_x.x.x.jar**
  - **org.apache.juneau.rest.server.jaxrs_x.x.x.jar**
  - **org.apache.juneau.rest.server_x.x.x.jar**
  - **org.apache.juneau.rest.client_x.x.x.jar**
  - **org.apache.juneau.microservice.jetty_x.x.x.jar**
  - **org.apache.juneau.microservice.core_x.x.x.jar**
  - **org.apache.juneau.marshall.rdf_x.x.x.jar**
  - **org.apache.juneau.marshall_x.x.x.jar**
  - **org.apache.juneau.dto_x.x.x.jar**
  - **org.apache.juneau.config_x.x.x.jar**
- **lib**
  - **juneau-svl-x.x.x.jar**
  - **juneau-rest-server-springboot-x.x.x.jar**
  - **juneau-rest-server-jaxrs-x.x.x.jar**
  - **juneau-rest-server-x.x.x.jar**
  - **juneau-rest-client-x.x.x.jar**
  - **juneau-microservice-jetty-x.x.x.jar**
  - **juneau-microservice-core-x.x.x.jar**
  - **juneau-marshall-rdf-x.x.x.jar**
  - **juneau-marshall-x.x.x.jar**
  - **juneau-dto-x.x.x.jar**
  - **juneau-config-x.x.x.jar**
- **juneau-all-x.x.x.jar**

Examine the contents of each inner jar file.  They should all have the following files with the correct content:

- **META-INF/LICENSE**
- **META-INF/NOTICE**

Each of the project zip files should be importable into Eclipse without compilation errors.  Try running the launchers to make sure the REST interfaces come up correctly.

## 6 - Dropping a release candidate

If a release candidate is to be dropped, e.g. it fails the **[VOTE]** or you cancel earlier, then:

In Nexus, tick the **staging repository** and **Drop**.

Delete the old tag:

```
git push origin :juneau-7.0.0-RC1
```

Roll back the SNAPSHOT version number:

```
mvn release:update-versions -DautoVersionSubmodules=true -DdevelopmentVersion=7.0.0-SNAPSHOT
```

In your dist checkout, **svn rm** the dropped RC folders:

```
svn rm *RC*
```

Raise JIRA issues for the reasons that caused the RC to be dropped.

## 7 - Distributing

Once a **[VOTE] [RESULT]** email has been agreed on **dev@juneau**, then:

- Tag the release in Git.
- Promote the release candidate to a release
- Update download page
- Send announcement email

## 7.1 - Tag the release in Git

Copy the final tag.

```
git tag juneau-7.0.0 juneau-7.0.0-RC1
git push --tags
```

## 7.2 - Moving to release area

Use full URL **svn mv** to move the accepted release candidate to the release folder structure on **dist.apache.org**, e.g. **https://dist.apache.org/repos/dist/release/juneau/source/** but remember to remove the -RC1 part from the folder name.

```
svn mv https://dist.apache.org/repos/dist/dev/juneau/source/juneau-7.0.0-RC1 https://dist.apache.org/repos/dist/release/juneau/source/juneau-7.0.0 -m "Releasing apache juneau-7.0.0"

svn mv https://dist.apache.org/repos/dist/dev/juneau/binaries/juneau-7.0.0-RC1 https://dist.apache.org/repos/dist/release/juneau/binaries/juneau-7.0.0 -m "Releasing apache juneau-7.0.0"
```

## 7.3 - Releasing Maven repository

Tick the correct **staging repository** (perhaps do a quick check of an SHA1 sum against the vote email), then click **Release**. It should then propagate to **Apache's release Maven repository** and eventually mirrored to **Maven Central**.

## 7.4 - Update download page

Wait 24 hours for the download mirrors to pick up the new release from **dist.apache.org**, otherwise eager users (or users who just happened to want to download that day) may get **404 Not Found** errors.

Update (or make) the corresponding pages under **downloads.html** including the correct version number for the Maven dependencies and the release dates.

Download links should be using the mirror redirector **https://www.apache.org/dyn/closer.cgi**, e.g. **https://www.apache.org/dyn/closer.cgi/juneau/source/juneau-7.0.0/juneau-7.0.0-source-release.zip**

The **asc/md5/sha1** links should go directly to **https://www.apache.org/dist/** - e.g. **https://www.apache.org/dist/juneau/source/juneau-7.0.0/juneau-7.0.0-source-release.zip.asc** (important bit: **https**)

Remove the staged release from **Nexus**.

## 7.5 - Removing old versions

**After** the download page has been published, the download mirrors have synchronized and the new version is live you must remove the old versions from **https://dist.apache.org/repos/dist/release/juneau/source/**

```
svn rm https://dist.apache.org/repos/dist/release/juneau/source/juneau-6.9.9/

svn rm https://dist.apache.org/repos/dist/release/juneau/binaries/juneau-6.9.9/
```

Older versions are archived under **http://archive.apache.org/dist/juneau/source/** but won't appear on the download mirrors.

## 7.6 - Create announcements

Announce the availability of the new release.

Please **remember to give a brief description of your component**.

**To: announce@apache.org, dev@juneau.apache.org**

**[ANNOUNCEMENT] Apache Juneau 9.x.x Released**

Hello all,

The Apache Juneau team is proud to announce the release of Juneau 9.x.x.

Apache Juneau is...

- A toolkit for marshalling POJOs to a variety of content types using a common framework.
- A REST server API for creating Swagger-based self-documenting REST interfaces using POJOs.
- A REST client API for interacting with REST interfaces using POJOs.
- A remote proxy API built on top of REST.
- A sophisticated INI config file API.
- A REST microservice API that combines all the features above for creating lightweight standalone REST interfaces that start up in milliseconds.

<A description of latest updates>

The release is available here (don't forget to verify the signatures):
xxx

Release notes can be found here:
xxx

We welcome your help and feedback. For more information on the project and how to get involved, visit the project website at
http://juneau.apache.org/

Send this mail from your @apache.org account.  Please spell check the document!

Wait to send the release announcement until you have verified that

1. The release artifacts are available on the mirrors.
2. The component website including the updated download page has been updated on the public site **http://juneau.apache.org/proper/juneau.**
3. If the component publishes maven artifacts, these artifacts have been replicated to the central maven repo at **repo.maven.apache.org**.
   (Clear your local repo of the release artifacts and either activate the clirr report with the updated version info or update a local project with a dependency to the new release version to get maven to try to download the release artifacts. Or just access repo using a web browser.)

## 8 - Updating Website

The website contents are located in GitHub:  https://github.com/apache/juneau-website

Checkout the contents of this repo to the same git folder of the juneau project (e.g. /juneau and /juneau-website should be in the same parent git folder).

Generate the Javadocs for the new version by doing the following:

1. Make sure **JUNEAU_VERSION** is set to the new version in **/juneau/juneau-env.sh**.  (should already be done at this point)
2. In the **juneau-doc/docs** folder, copy-and-replace all instance of "**<juneauVersion>last-version</juneauVersion>**" with "**<juneauVersion>current-version</juneauVersion>**" (e.g. "**<juneauVersion>9.0.0</juneauVersion>**" with "**<juneauVersion>9.0.1</juneauVersion>**").
3. In the **juneau-doc/docs** folder, copy-and-replace all instance of "**<juneauVersionNext>current-version</juneauVersionNext>**" with "**<juneauVersionNext>next-version</juneauVersionNext>**" (e.g. "**<juneauVersionNext>9.0.1</juneauVersionNext>**" with "**<juneauVersionNext>9.0.2</juneauVersionNext>**".
4. Run the following command in the **/juneau** folder to generate the Javadocs and add them to the website:  **./juneau-build-javadoc.sh**
5. Verify that a new folder is created at **juneau-website/content/site/apidocs-x.x.x** containing the newly-generated javadocs.

Update the downloads page and regenerate templates.

1. Update the **juneau-website/templates/downloads.html** to reflect the newest release.
   a. Change **Current release** section.
   b. Add new entry to **Older releases** section.
2. Update the settings in **juneau-website.properties** to reflect new versions.
3. Run the **juneau-website.sh** script to regenerate the templated files.

After completion, commit and push the changes to the master branch.  The website should be updated immediately.