

Release a hotfix

In this tutorial we will describe how to release a hotfix.

01 - Prerequisites

- We need clone of git [repository](#) with checkout actual branch with hotfix, during the process, we will need the **right to write** to this repository, example:

```
# clone repository
$ git clone git@github.com:bcvsolutions/CzechIdMng.git
# change folder
$ cd CzechIdMng
# checkout hotfix
$ git checkout hotfix/<HOTFIX_VERSION>
```

- install maven to your computer ([maven install steps](#)),

Hotfix branch not exists

If branch with hotfix doesn't exist, you can easily create this branch with this command (you must be on branch **master** and locally or remote must not exist branch with name hotfix...):

```
$ cd Realization/backend/aggregator/
$ mvn clean -Prelease jgitflow:hotfix-start
$ git push -u origin hotfix/<HOTFIX_VERSION>:hotfix/<HOTFIX_VERSION>
```

This command creates new hotfix started from **master** branch. Sometimes it is needed to fix previously released version. For this case, we can find last commit id (= last commit in tag) and start hotfix with command:

```
$ mvn clean -Prelease jgitflow:hotfix-start -
DstartCommit=<CHANGE_WITH_COMMIT_ID> -
DreleaseVersion=<CHANGE_WITH_CURRENT_HOTFIX_VERSION_WITHOUT_SNAPSHOT_SUFFIX>
```

Manually clean/fetch local branches

Execute these three commands in CzechIdM folder.

```
$ git fetch -p
$ git branch -vv
$ git fetch -p && for branch in `git branch -vv | grep ': gone]' | awk
'{print $1}'`; do git branch -D $branch; done
```

This command creates new branch and sets up new version in BE poms. *FE package.json* you must

update by self.

02 - Maven settings

After install maven is necessary setup maven via **settings.xml**, this file you can found in **~/.m2/**. If this file don't exists create it. Into file you must put this settings (change username and password yours!):

```
<settings>
  <servers>
    <!-- Nexus server login information -->
    <server>
      <id>nexus</id>
      <username><USERNAME></username>
      <password><PASSWORD></password>
    </server>
  </servers>

  <pluginGroups>
    <pluginGroup>external.atlassian.jgitflow</pluginGroup>
  </pluginGroups>

  <mirrors>
    <mirror>
      <id>nexus</id>
      <mirrorOf>*</mirrorOf>
      <url>https://nexus.bcvolutions.eu/repository/maven-public/</url>
    </mirror>
  </mirrors>

  <profiles>
    <profile>
      <id>nexus-repo</id>
      <repositories>
        <repository>
          <id>maven-snapshots</id>
          <url>https://nexus.bcvolutions.eu/repository/maven-
snapshots/</url>
          <releases><enabled>false</enabled></releases>
          <snapshots><enabled>true</enabled></snapshots>
        </repository>
        <repository>
          <id>maven-release</id>
          <url>https://nexus.bcvolutions.eu/repository/maven-
releases/</url>
          <releases><enabled>true</enabled></releases>
          <snapshots><enabled>false</enabled></snapshots>
        </repository>
      </repositories>
    </profile>
  </profiles>
</settings>
```

```
</profile>
</profiles>
<activeProfiles>
  <activeProfile>nexus-repo</activeProfile>
</activeProfiles>
</settings>
```

03 - Update FE package.json with actual version

- Update version in FE package.json **is not** automatic. You must update this information by self [example](#). This changes you must commit and push before start with release.

04 - Release via aggregator

Follow the following commands:

```
# in your local clone of repository go to
$ cd CzechIdMng/Realization/backend/aggregator
# start with release hotfix
$ mvn clean -Prelease jgitflow:hotfix-finish -DupdateDependencies=false -
DdocumentationOnly=true -DallowUntracked=true -
DdevelopmentVersion=<CHANGE_WITH_CURRENT_DEVELOPMENT_BRANCH>
```



If you have access to GitHub via username and password (not ssh key), you will be asked for this information during the release

If you get during process this error: The authenticity of host 'git.bcvolutions.eu' can't be established. add into parameters this:

```
$ mvn ... -DenableSshAgent=true
```

If you have untracked files which will definitely not cause GIT conflicts, add the following parameter:

```
$ mvn ... -DallowUntracked=true
```

- After mvn finish with build, test and update just enter this on hotfix branch:

```
$ git push
$ git push --tags
```



If hotfix is started from master branch, don't forget to merge your branch (= latest release) into master.



If hotfix is not started from master branch, then after finish with hotfix, merge hotfix



branch in develop only, **not to master**. Sometimes is development version to different and automatic merge fails anyway. If old version is hotfixed, then most probably hotfix-finish command fails, after artifacts are deployed to nexus and when master branch is updated - **this is ok** - we don't want to merge "old" version into master. Just ignore changes on master branch, create tag on hotfix branch manually (e.g. directly from github) a merge hotfix branch into develop manually. After merge, you can delete hotfix branch.



After success release don't forget to delete hotfix branch.

From:

<https://wiki.czechidm.com/> - CzechIdM Identity Manager

Permanent link:

https://wiki.czechidm.com/tutorial/dev/how_to_hotfix

Last update: **2017/11/28 12:19**

