Git sustav za praÄ enje promjena koji koristi Koha.

Mnogo bolje i potpunije upute: http://wiki.koha-community.org/wiki/Version_Control_Using_Git

DON'T PANIC!

Contents: [Koha Croatian user group]

podaci o korisniku

Da bi vaÅ¡i commiti imali lijepo ime korisnika, morate prvo konfigurirati git

```
$ git config --global user.name "Dobrica Pavlinusic"
$ git config --global user.email "dpavlin@rot13.org"
```

Ovo je potrebno napraviti **samo jednom** za korisnika i primjenjuje se na svim repozitorijima na istoj maÅ¡ini.

repozitoriji

Repozitoriji su jednostavno serveri koji nam omoguÄ avaju da na njih Å¡aljemo (push) ili sa njih vuÄ emo (pull) promjene koje su u njima napravljene.

Kako je svaki checkout git-a potpuni repozitorij, svaki od njih moÅ¾e postati javni repozitorij koji moÅ¾e dijeliti promjene.

Upstream Koha development

- http://git.koha.org/
- git://git.koha.org/pub/scm/koha.git

Checkout (lokalna kopija)

```
git clone git://git.koha.org/pub/scm/koha.git
```

FFZG promjene

- http://git.rot13.org/?p=koha.git;a=summary
- git://git.rot13.org/koha.git
- ssh://git.rot13.org/git/koha/

promjene iz drugog repozitorija

Ako Å¾elimo povuÄ i promjene iz drugog repozitorija (npr. druge instalacije unutar KOHA CUG-a) trebamo dodati novi remote repozitorij.

Na repozitoriju koji je napravljen od upstream Koha repozitorija, imamo jedan remote repozitorij `origin` za push i pull:

```
dpavlin@t61p:/srv/koha$ git remote -v
origin  git://git.koha.org/pub/scm/koha.git (fetch)
origin  git://git.koha.org/pub/scm/koha.git (push)
```

Sada Ä emo dodati repozitorij sa FFZG development-a koristeÄ i obiÄ nu ssh konekciju:

```
dpavlin@t61p:/srv/koha$ git remote add ffzg ssh://koha-dev.rot13.org/srv/koha/
dpavlin@t61p:/srv/koha$ git remote -v
ffzg    ssh://koha-dev.rot13.org/srv/koha/ (fetch)
ffzg    ssh://koha-dev.rot13.org/srv/koha/ (push)
origin  git://git.koha.org/pub/scm/koha.git (fetch)
origin  git://git.koha.org/pub/scm/koha.git (push)
```

i povuÄ i sve promjene lokalno:

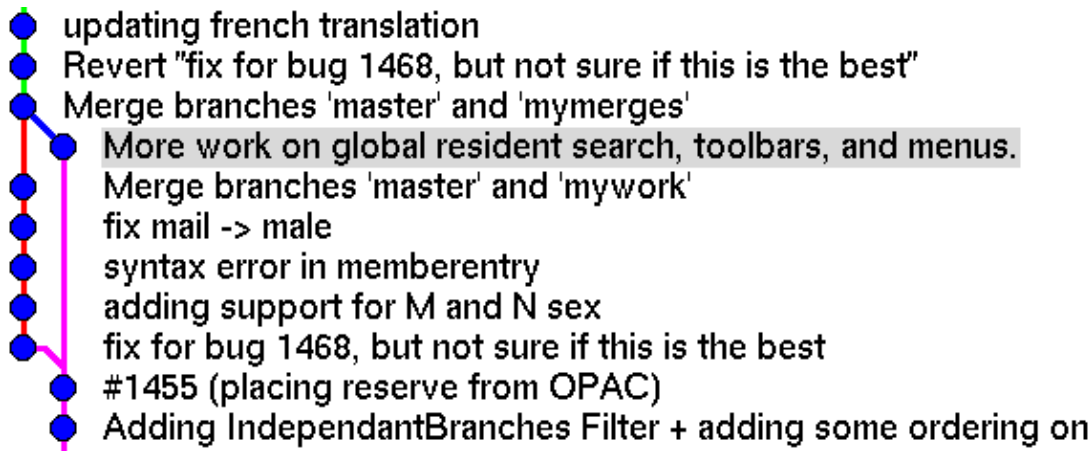
```
dpavlin@t61p:/srv/koha$ git fetch ffzg
remote: Counting objects: 2019, done.
remote: Compressing objects: 100% (1440/1440), done.
remote: Total 1790 (delta 1116), reused 587 (delta 317)
Receiving objects: 100% (1790/1790), 1.40 MiB | 1.12 MiB/s, done.
Resolving deltas: 100% (1116/1116), completed with 67 local objects.
From ssh://koha-dev.rot13.org/srv/koha
 * [new branch]      ffzg       -> ffzg/ffzg
 * [new branch]      ffzg2      -> ffzg/ffzg2
 * [new branch]      koha-lxc   -> ffzg/koha-lxc
 * [new branch]      master     -> ffzg/master
 * [new branch]      origin     -> ffzg/origin
```

```
 * [new branch]      rt-390-signatura-zatvorenog-spremista -> ffzg/rt-390-signatura-zatvorenog-sp
From ssh://koha-dev.rot13.org/srv/koha
 * [new tag]         ffzg       -> ffzg
```

branches

Branches su mjesta na kojima se povjest promjena grana u dva dijela. Jedan je development koji rade upstream developeri Kohe, a drugi su naÅ¡e lokalne promjene za neÅ¡to.

Nakon nekog vremena, potrebno je dva branch-a spojiti u jedan da bi preuzeli promjene. `gitk` nam moÅ¾e pokazati to kao stablo:



Promjene se mogu preuzeti (merge) iz jednog branch-a u drugi.

imenovanje

Branches nisu niÅ¡ta drugo nego ime za svaku toÄ ku i kojoj se razvoj dijeli u dva toka.

Dobri nazivi za branch:

- DNS hostname produkcijske ili develpment maÅ¡ine na kojoj je branch checkoutan
- veza na bug tracking (prefix-broj-kratak-opis)

pregled

```
dpavlin@t61p:/srv/koha$ git branch -a
* master
  remotes/ffzg/ffzg
  remotes/ffzg/ffzg2
  remotes/ffzg/koha-lxc
  remotes/ffzg/master
  remotes/ffzg/origin
  remotes/ffzg/rt-390-signatura-zatvorenog-spremista
  remotes/origin/3.0.x
  remotes/origin/HEAD -> origin/master
  remotes/origin/biblibre-acq-preview-only
  remotes/origin/biblibre-integration
  remotes/origin/biblibre-sopac
  remotes/origin/labels_recon
```

```
  remotes/origin/master
  remotes/origin/rfid-direct-tagging
  remotes/origin/sysprefs_editor
```

kreiranje

Pretpostavimo da Å¾elimo kreirati branch za instalaciju na `koha.ffzg.hr`:

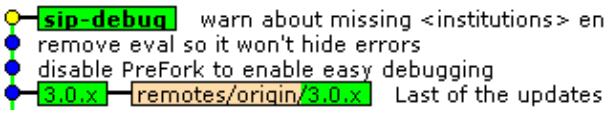
```
dpavlin@t61p:/srv/koha$ git checkout -b koha.ffzg.hr remotes/ffzg/ffzg2
Branch koha.ffzg.hr set up to track remote branch ffzg2 from ffzg.
Switched to a new branch 'koha.ffzg.hr'
```

patches

format-patch

<u>git format-patch</u>

NaÅ¡ cilj je napraviti patcheve od promjena koje smo napravili u naÅ¡em branch-u:



Iz slike vidimo da smo napravili `sip-debug` branch iz postojeÄ eg `3.0.x`

```
koha-lxc:/srv/koha# git format-patch -o sip2-changes 3.0.x
sip2-changes/0001-disable-PreFork-to-enable-easy-debugging.patch
sip2-changes/0002-remove-eval-so-it-won-t-hide-errors.patch
sip2-changes/0003-warn-about-missing-institutions-entry-in-config-xm.patch
```

Ovo Ä e napraviti onoliko patcheva koliko imamo commita na naÅ¡em branch-u koji smo napravili od `3.0.x`

am (apply mail patch)

Prvo Ä emo napraviti novi branch u naÅ¡em repozitoriju `sip2-debug` u koji Ä emo importati naÅ¡e patcheve:

```
koha-lxc:/srv/koha# git checkout -b sip2-debug master
Switched to a new branch "sip2-debug"
```

Nakon toga moÅ¾emo applyati samo one promjene koje su nam interesantne:

```
koha-lxc:/srv/koha# git am sip2-changes/0001-disable-PreFork-to-enable-easy-debugging.patch
```

```
Applying disable PreFork to enable easy debugging

koha-lxc:/srv/koha# git am sip2-changes/0002-remove-eval-so-it-won-t-hide-errors.patch
Applying remove eval so it won't hide errors

koha-lxc:/srv/koha# git am sip2-changes/0003-warn-about-missing-institutions-entry-in-config-xm.p
Applying warn about missing <institutions> entry in config xml
```

slanje na koha-patches listu

Koha prima patcheve preko koha-patches liste

yada, yada, nije testirano!

```
# create all changes against master in patches dir:
git format-patch -p patches master
git send-mail --to koha-patches@koha.org patches

# reciving side would just pull whole mbox
git am mbox
```

ne Å¾elim viÅ¡e svoje lokalne promjene!

```
git reset --hard
```

Oprez ovo Ä e pobrisati **sve promjene koje su razliÄ ite od zadnjeg commit-a**

ViÅ¡e informacija

- RailsConf Git Talk (mnogo zgodnih animacija Å¡to se deÅ¡ava sa pojedinim komandama)
- Koha wiki ima upute za koriÅ¡tenje git-a

Javni repozitorij

Da bi drugi korisnici mogli kopirati promjene, dovoljan je samo ssh. MeÄ utim, za anonimne korisnike treba podesiti git-daemon

Dobar savjet je **prvo** instalirati gitweb za overview.

git-daemon

```
git-daemon --verbose --export-all --base-path=/var/cache/git
```

U Debian-u postoji paket `git-daemon-run` koji koristi runit (YMMV) koji se podeÅ¡ava u

```
dpavlin@mjesec:~$ cat /etc/sv/git-daemon/run
```

```
#!/bin/sh
exec 2>&1
echo 'git-daemon starting.'
exec git-daemon --verbose --export-all --base-path=/var/cache/git
```

Dodan je `--export-all` i popravljen `--base-path` (sic!)