

# Git hooks

Leon Samardžić, Marin Frankić, Viktor Antunović

Preddiplomski studij računarstva, Tehnički fakultet, Sveučilište u Rijeci

23.1.2018

## Što su Git Hooks?

- ▶ Git Hooks su skripte koje Git izvrava prije ili poslije događaja kao to su: commit, push i recieve. Git Hooks su ugradena znacajka - nema potrebe za preuzimanjem. Git Hooks se izvode na lokalnoj razini.
- ▶ Ove skripte ogranicene su samo matom razvojnog programera.
- ▶ Jezik skripte odreduje shebang (shebang je znakovna sekvencija koja se sastoji od znaka ljestv i usklicnika (!) na pocetku skripte.) Zapisnik kao to je to obicno u Linux softveru. Imajte na umu da se to odnosi i na Windows jer Git za Windows radi pod MSYS (MSYS je zbirka GNU-ovih uslunih programa kao to su bash, make, gawk i grep kako bi se omogucilo stvaranje aplikacija i programa koji ovise o tradicionalno UNIX alatima. Namijenjen je za dopunu MinGW i nedostataka cmd shell-a.).
- ▶ Git Hooks nalaze se ispod `.git/hooks/direktorija` vaeg spremitea.

# TYPES OF HOOKS

- ▶ Committing-Workflow Hooks
- ▶ E-mail Workflow Hooks
- ▶ Server-Side Hooks
- ▶ Other Client Hooks

## applypatch-msg

- ▶ Ovaj hook se poziva s 'git am'. Potrebno je jedno parametar, ime datoteke koja sadri predloenu stavku log poruku.
- ▶ Ovom hook-u je dopusteno da mijenja poruku file-a na mjestu, i moze se iskoristiti da se normalizira poruka u neki standardni format.
- ▶ Također se može koristiti za odbijanje commit-a nakon pregleda datoteku poruke.

## pre-applypatch

- ▶ Ovaj hook se poziva s 'git am'. Ne treba nikakav parametar, i pozvana je nakon primjene zakrpa, ali prije nego to se izvrši commit.
- ▶ Ako izlazi s statusom koji nije nula, tada stablo za rad neće biti commitano nakon primjene zakrpa.
- ▶ Može se koristiti za pregledavanje radnog stabla i odbijanje izvravanja commit-a ako ne prođe određeni test.
- ▶ Zadana postavka "pre-applypatch", kada je omogućena, pokreće "pre-commit" hook, ako je ovo omogućeno.

## post-apppatch

- ▶ Ovaj hook se poziva s 'git am'. Ne treba nikakav parametar, i poziva se nakon primjene zakrpe i commit-a.
- ▶ Taj hook se prvenstveno namijenjuje za obavijesti i ne moe utjecati na rezultat 'git am'.

## pre-commit

- ▶ Ovaj hook se zove "git commit", a moe se zaobici s opcijom '-no-verify'. Nema nikakvih parametara.
- ▶ Izlazak s statusom koji nije nula iz ove skripte uzrokuje da se naredba "git commit" prekine prije stvaranja obveze.
- ▶ Sve hooks "git commit" se pozivaju na environment varijabla "GIT\_EDITOR =:  
" *akonaredbanepojaviuredivaczaizmjenuporukeobveze.*

## prepare-commit-msg

- ▶ Ovaj hook se poziva s 'git commit' odmah nakon pripreme zadane poruke zapisnika i prije pokretanja uredivaca.
- ▶ Potrebno je jedan do tri parametra.
  1. naziv datoteke koji sadri poruku log zapisa.
  2. izvor commit poruke- moze biti "message", "template", "merge", "squash" i "commit"
- ▶ Ako izlazni status nije nula, "git commit" ce se prekinuti.
- ▶ Svrha hook-a je urediti poruku datoteke na mjestu, i to nije potisnuto opcijom "--no-verify". Izlaz koji nije nula znaci neuspjeh hook-a i prekid commit-a. Ne bi se trebalo koristiti kao zamjena za pre-commit hook.



## commit-msg

- ▶ Ovaj se hook poziva 'git commit' i 'git merge', i to moe biti zaobideno s opcijom "--no-verify".
- ▶ Potreban je samo jedan parametar:
  - ▶ naziv datoteke koja sadri predloenu poruku log zapisa.
- ▶ Izlazak s statusom koji nije nula uzrokuje naredbu za prekid.
- ▶ Hook-u je doputeno uredivanje datoteke poruke na mjestu.
- ▶ Zadana "commit-msg" kuka, kada je omogucena, detektira duplikate "Signed-off-by" linija i prekida commit ako je pronadena jedna.

## post-commit

- ▶ Ovaj hook se poziva s 'git commit'. Ne treba nikakvih parametara i pozvan je nakon izvršenja commit-a.
- ▶ Taj je hook prvenstveno namijenjen obavijestima i ne moe utjecati na ishod "git commita".

## pre-rebase

- ▶ Ovaj hook naziva se 'git rebase' i moe se koristiti za sprjecavanje grane od re-basea. Kuka se moe pozvati s jednim ili dva parametra.

## post-checkout

- ▶ Ovaj se hook poziva kada se pokrene git checkout nakon auranja radnog stabla.
- ▶ Parametri:
  - ▶ ref prethodnog HEAD-a
  - ▶ ref novog HEAD-a koji moe, ali i ne mora biti promijenjen
  - ▶ flag koji pokazuje je li checkout bio brach checkout ili file checkout
- ▶ Ne utjece na ishod checkouts.

## post-checkout

- ▶ Također se pokreće nakon git clone naredbe, osim ako se koristi opcija `-no-checkout` (`-n`).
- ▶ Parametri:
  - ▶ `null-ref`
  - ▶ `ref` novog HEAD-a i `flag` su uvijek 1 kao i za git worktree add naredbu ako se ne koristi `-no-checkout`
- ▶ Ovaj se hook može koristiti za provođenje provjere valjanosti repozitorija, automatsko prikazivanje razlika od prethodnog HEAD-a ako su različiti ili za postavljanje postavki radnog direktorija.

## post-merge

- ▶ Ovaj se hook poziva kada se pokrene git merge to se događa kada se obavi git pull na lokalnom repozitoriju.
- ▶ Ima jedan parametar, statusni flag koji specifikira je li ili nije merge koji je obavljen squash merge.
- ▶ Ne utječe na ishod naredbe git merge i ne izvrava se ako nastanu konflikti.
- ▶ Moe se koristiti s odgovarajucim pre-commit hookom za spremanje i vraćanje bilo kojeg oblika metapodataka povezanih s radnim stablom npr. doputenja, vlasnitvo itd.

## pre-push

- ▶ Ovaj se hook poziva kada se pokrene naredba `git push` i može se koristiti za prekidanje izvravanja.
- ▶ Poziva se s dva parametra koji daju ime i lokaciju remote-a, ako remote nije koriten obje vrijednosti će biti jednake.
- ▶ Informacije o tome što treba biti pushano:
- ▶ Na primjer ako se pokrene naredba `"git push origin master:foreign"` hook bi dobio ovu liniju:
  - ▶ `refs/heads/master 67890 refs/heads/foreign 12345`
- ▶ Ako ovaj hook izlazi sa statusom koji nije nula, `git push` će se prekinuti bez izvravanja.

## pre-receive

- ▶ Ovaj se hook poziva s `git-receive-pack` naredbom kada reagira na `"git push"` i aurira referencu na repozitorij.
- ▶ Poziva se prije auriranja refova na remote repozitoriju. Njegov izlazni status određuje je li auriranje uspješno.
- ▶ Izvrava se jednom za receive operaciju. Nema argumenata, ali za svaki ref koji se aurira dobija liniju:
- ▶ Ako hook izlazi sa statusom koji nije nula, refovi neće biti aurirani.
- ▶ Ako hook izade sa statusom nula, auriranje individualnih refova još uvijek može biti zaustavljeno s `update hook-om`.



## update

- ▶ Ovaj se hook poziva s naredbom `git-receive-pack` kada reagira na "git push" i aurira reference u repozitoriju.
- ▶ Poziva se prije auriranja refova na remote repozitoriju. Njegov izlazni status određuje je li auriranje uspješno.
- ▶ Hook se izvršava jednom za svaki ref koji se mora aurirati i uzima tri parametra:
  - ▶ Ime refa koji se aurira
  - ▶ Stari object name u refu
  - ▶ Novi object name koji će se spremiti u ref

## update

- ▶ Ako je izlazni status nula, hook doputa auriranje.
- ▶ Ako je izlazni status nije nula onda se prekida auriranje.
- ▶ Ovaj se hook moe koristiti da bi se prekinulo forsirano auriranje.
- ▶ U okruenju koje ogranicava priatuo korisnika samo za git naredbe preko ice, ovaj se hook moe koristiti za
- ▶ implementaciju kontrole pristupa bez oslanjanja na vlasnitvo nad datotekama.

## post-receive

- ▶ Ovaj se hook poziva s "git-receive-pack" naredbom kada reagira na "git push" i aurira reference u repozitoriju.
- ▶ Izvrava se na remote repozitoriju nakon to su svi refovi aurirani.
- ▶ Izvrava se jednom za operaciju receive. Ne uzima argumente.
- ▶ Ne utjece na ishod "git-receive-pack" jer je pozvan nakon to je pravi posao gotov.
- ▶ Zadani post-receive hook je prazan, ali postoji primjerak skripte "post-receive-email" naveden u contrib/hooks direktoriju.

## post-update

- ▶ Ovaj se hook poziva s "git-receive-pack" naredbom kada regira na "git push" i aurira reference u repozitoriju.
- ▶ Izvrava se na remote repozitoriju nakon to su svi refovi aurirani.
- ▶ Potreban je promjenjivi broj parametara, od kojih je svaki naziv refa koji je auriran.
- ▶ Ovaj je hook prvenstveno namijenjen obavijestima i ne moe utjecati na ishod "git-receive-pack" naredbe.

## push-to-checkout

- ▶ Ovaj se hook poziva s "git-receive-pack" naredbom kada regira na "git push" i aurira reference u repozitoriju a kada push pokua aurirati granu koja je trenutno "checked out" i "receive.denyCurrentBranch" konfiguracijska varijabla postavljena na "updateInstead". Takav je push nedoputen ako radno stablo i indeks remote repozitorija imaju razlike u odnosu na trenutni "checked out" commit. Ovaj hook koristi se za prelaenje preko tog zadanog ponaanja.
- ▶ Hook prima commit s kojim ce se vrh trenutne grane aurirati.
- ▶ Moe izaci sa statusom koji nije nula da odbije push ili moe napraviti potrebne promjene na radnom stablu i indeksu da ih dovede u eljeno stanje kada je vrh trenutnog stabla auriran na novi commit, i izaci sa statusom nula.

## pre-auto-gc

- ▶ Ovaj se hook poziva s "git gc -auto" naredbom. Ne treba nikakav parametar i izlazak sa statusom koji nije nula
- ▶ Uzrokuje prekid naredbe.

## post-rewrite

- ▶ Ovaj se hook poziva s naredbama koji prepisuju commitove. Prvi argument označava naredbu kojom je pozvan.
- ▶ Hook dobiva listu prepisanih commitova u formatu:
- ▶ Hook se uvijek pokreće nakon to se dogodi automatsko kopiranje biljeki i stoga ima pristup tim biljekama.

## sendemail-validate

- ▶ Ovaj se hook poziva s naredbom "git send-email". Potreban je jedan parametar, ime datoteke koja sadri email koji treba biti poslan.
- ▶ Izlaskom sa statusom koji nije nula "git send-email" se prekida prije slanja emailova.



## fsmonitor-watchman

- ▶ Ovaj se hook poziva kada je konfiguracijska opcija `core.fsmonitor` poslana u `.git/hooks/fsmonitor-watchman`.
- ▶ Uzima dva argumenta, verziju (trenutno 1) i vrijeme u proteklim nanosekundama od ponoci, 1. siječnja 1970.
- ▶ Hook bi trebao poslati listu datoteka u radnom direktoriju koji su se promijenili od traenog vremena.
- ▶ Git će ograniciti koje datoteke pregledava kao i koje direktorije pregledava.
- ▶ Izlazni status određuje hoće li git koristiti podatke iz hooka da ograniči pretraivanje.
- ▶ U slučaju greke vratiti će se na verificiranje svih datoteka i foldera.