Github

SADRŽAJ

[1. HISTORIJAT 1](#_Toc510801834)

[1.1. Github kroz vrijeme 1](#_Toc510801835)

[2. GIT 2](#_Toc510801836)

[2.1 Preuzimanje Git-a 2](#_Toc510801837)

[3. OSNOVNE KOMANDE 3](#_Toc510801838)

[3.1. Novi repozitorij 3](#_Toc510801839)

[3.2. Checkout repozitorija 3](#_Toc510801840)

[3.3. Workflow 3](#_Toc510801841)

[3.4. Add & Commit 4](#_Toc510801842)

[3.5. Pushing 4](#_Toc510801843)

[3.6. Grananje 4](#_Toc510801844)

[3.7. Update & Merge 5](#_Toc510801845)

[3.8. Pregled repozitorija 6](#_Toc510801846)

[3.9. Zamjena lokalnih promjena 6](#_Toc510801847)

# 1. HISTORIJAT

Razvoj GitHub platforme je počeo 1. oktobra 2007. Sajt je pokrenut u aprilu 2008. od strane Tom Preston-Vernera, Krisa Vanstrata i P.J. Hajeta, par mjeseci pošto je bio dostupan u beta izdanju. Projektima na GitHub-u se pristupa i manipuliše koristeći standardni Git interfejs iz komandne linije preko kojeg su dostupne i sve standardne Git komande. GitHub dopušta registrovanim i neregistrovanim korisnicima da pregledaju javna skladišta na sajtu. Nekoliko desktop klijenata i Git priključaka su kreirani pomoću GitHub-a koji se integrišu sa platformom. GitHub sajt pruža funkcije slične društvenim mrežama kao što su: dovodi (engl. feeds), pratioci (engl. followers), viki (koristeći softver Gollum) i grafikone saradnje koji ukazuju kako programeri rade na svojim verzijama ("račvama") skladišta i koja račva (i grana te račve) je najnovija. Korisnik mora da napravi nalog kako bi doprinjeo stranici, ali javna skladišta mogu biti pregledana i preuzeta od strane bilo koga. Sa registrovanim korisničkim nalogom, korisnici mogu da diskutuju, upravljaju skladištima, prave nova skladišta, postavljaju doprinose drugim skladištima i pregledaju izmene u kodu. Softver koji pokreće GitHub je pisan koristeći Ruby on Rails i Erlang od strane GitHub, Inc. razvojnog tima: Kris Vanstrat, P.J. Hajet i Tom Preston-Verner.

# 1.1. Github kroz vrijeme

24. februar 2009: Članovi GitHub tima su najavili, u razgovoru u sjedištu Yahoo!-a, da je prve godine na mreži, GitHub sakupio preko 46,000 javnih skladišta od kojih je 17,000 formirano samo u prethodnom mesecu. U to vreme, oko 6,200 skladišta je odvojeno barem jednom, a 4,600 je spojeno.

5. jul 2009: GitHub je objavio da sajt sada koristi više od 100,000 korisnika.

27. jul 2009: U drugom razgovoru održanom u Yahoo!-u, Tom Preston Verner objavio je da je GitHub porastao da bude host za 90,000 jedinstvenih javnih skladišta, 12,000 što se odvaja barem jednom, za ukupno 135,000 skladišta.

25. jul 2010: GitHub je saopštio da je host 1 miliona skladišta.

20. april 2011: GitHub je saopštio da je host 2 miliona skladišta.

2. juni 2011: ReadWriteWeb je izvestio da je GitHub prevazišao SourceForge i Google Code u ukupnom broju urezivanja za period od januara do maja 2011. godine.

9. jul 2012: Piter Levin, generalni partner GitHub-a Andresen Horvic, naveo je da prihodi od GitHub-a rastu 300% godišnje od 2008. godine "profitabilan skoro ceo put".

16. januar 2013: GitHub je objavio da je prešao 3 miliona označenih korisnika i tada je bio host više od 5 miliona skladišta.

23. decembar 2013: GitHub je objavio da je dostigao 10 miliona skladišta.

U junu 2015. GitHub je otvorio kancelariju u Japanu i to im je prva kancelarija izvan SAD -a.

29. jul 2015: GitHub je objavio da je podigao 250 miliona dolara iz fonda Sequoia Capital. Zaokružena je vrijednost kompanije na približno 2 milijarde dolara.

# 2. GIT

Git je distribuirani sustav za upravljanje izvornim kodom nastao 2005. godine iz nezadovoljstva razvijatelja Linux operacijskog sustava karakteristikama postojećeg sustava za upravljanje izvornim kodom ("BitKeeper"). Git je dostupan na Linuxu, Windowsu i MacOS-u

## 2.1 Preuzimanje Git-a

Preuzimanje Git-a za MacOS se obavlja na linku : <https://git-scm.com/download/mac>

Preuzmanje Git-a za Windows se obalja na linku : <https://gitforwindows.org/>

Preuzimanje Git-a na Linux (Fedora) se vrši komandom :

$ sudo dnf install git-all

Preuzimanje Git-a na Linux (Debian) se vrši komandom :

$ sudo apt install git-all

# 3. OSNOVNE KOMANDE

## 3.1. Novi repozitorij

Da bi ste kreirali novi repozitorij morate napraviti novi direktorij, koji će te otvoriti i izvršiti komandu

**git init**

da bi ste kreirali novi git repozitorij.

## 3.2. Checkout repozitorija

Za kreiranje kopije vašeg lokalnog repozitorija koristimo komandu :

**git clone /path/to/repository**

**Kada koristimo udaljeni server, vaša komanda će biti :**

**git clone username@host:/path/to/repository**

## **3.3. Workflow**

Vaš lokalni repozitorij se sastoji od tri dijela koja su održavana od strane Gita. Prvi je „Working Directory“ koji sadrži stvarne fajlove, drugi je „Index“ koji je zapravo „staging area“ (područje gdje se nalaze podaci koje želimo dodati) i treći je „HEAD“ koji ukazuje na posljedni commit koji ste uradili.



## 3.4. Add & Commit

Da bi ste svoje promjene spremili na git potrebno je najprije da ih dodate u „Index“ koristeći komande :

git add <filename>

git add \* (\* označava sve fajlove unutar direktorija)

Vaši su fajlovi trenutno u „Indexu“ da bi ste ih spremili u „HEAD“ potrebno je izvršiti „commit“ sa komandom :

git commit -m "Commit message"

Sada su vaši fajlovi u „HEAD“ ali nisu još u vašem repozitoriju na githubu.

## 3.5. Pushing

Sada su vaše izmjene u „HEAD“ vašeg lokalnog repozitorija. Da bi ste vaše promjene spremili u repozitorij na githubu potrebno je izvršiti komandu :

git push origin master

Za drugu granu githuba (Grananje – chapter 3.6) izmjenite „master“ u naziv željene grane.

Ukoliko niste klonirali sadašnji repozitorij i želite ga spojiti sa vaš repozitorij na server potrebno je izvršiti komandu:

git remote add origin <server>

Sada ste u mogućnosti izvršiti „push“.

## 3.6. Grananje

Grane služe da bi se direktoriji ili fajlovi odvojili jedni od drugih. „Master“ grana je „default“ grana kada kreirate repozitorij. Koristite ostale grane za razvoj i naknadno ih spojite kada ste obavili svoj zadatak.



Kreirajte novu granu pod nazivom „feature\_x“ komandom:

git branch feature\_x

Da bi ste prešli iz grane master u granu „feature\_x“ morate izvršiti komandu:

git checkout feature\_x

Da bi ste obrisali granu morate unijeti komandu:

git branch -d feature\_x

Grana nije dostupna ako niste obavili „push“ te grane na vaš repozitorij na githubu. To se radi preko komande :

git push origin <branch>

## 3.7. Update & Merge

Da bi ste updejtovali vaš lokalni repozitorij sa novijim „commit“-om izvršite komandu

git pull

u vašem lokalnom direktoriju da bi ste dobavili (fetch) i spojili (merge) promjene.

Da bi ste spojili neku granu sa trenutnom koristite komandu

**git merge <branch>**

u bilo kojem slućaju git pokušava automatski spojiti promjene. Nažalost ovo nije uvijek moguće i rezultira u konfliktima. Vi ste odgovorni spojiti ove konflikte manualno tako što će te editovati fajlove prikazane u gitu. Nakon promjena morate označiti fajlove kao spojene sa komandom

git add <filename>

Prije spajanja promjena možete i napraviti pregled promjena komandom :

git diff <source\_branch> <target\_branch>

## 3.8. Pregled repozitorija

U najjednostavnijoj formi možete pregledati historiju repozitorija komandom

git log

Možete dodavati i parametre vaštoj pretragi sa komandama :

git log --author=bob

za pregled „commita“ čiji je autor Bob.

Ovo je samo jedan parametar. Za više parametara možete imati potpun pregled komandi unosom:

git log –help

## 3.9. Zamjena lokalnih promjena

U slućaju da ste uradili nešto pogrešno možete zamjeniti vaše lokalne promjene sa onima iz posljednjih koje se nalaze u „HEAD“. Promjene koje su dodane u „Index“ kao i novi fajlovi će biti zadržani. Komanda za zamjenu lokanlih promjena je :

git checkout -- <filename>

Ukoliko želite sve vaše promjene i lokalne „commit“-e izbrisati, morate dobaviti posljednju historiju sa servera i usmjeriti vašu lokalnu granu sa komandama

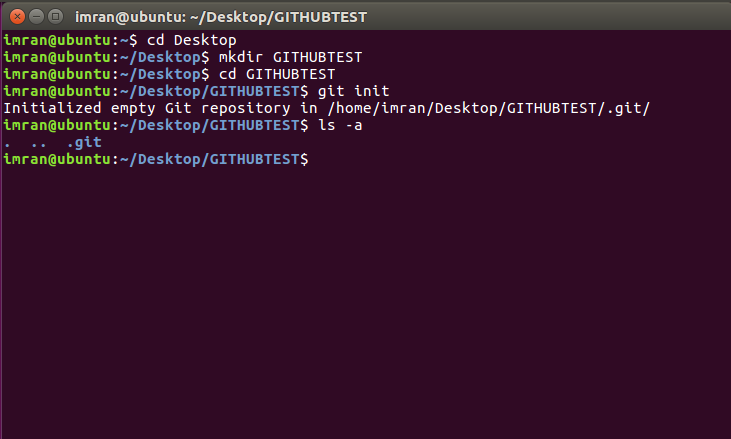
git fetch origin

git reset –hard origin/master

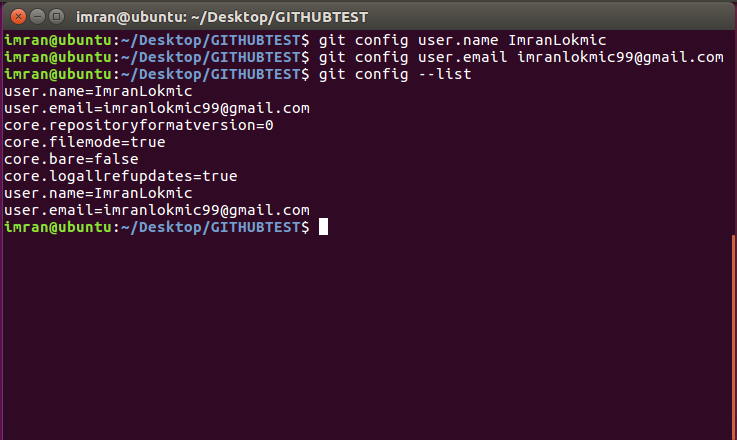
# 4. PRIMJER

Da bi se bolje objasnio princip rada u Git-u objasniti ćemo sljedeći primjer stvaranja tekstualnog fajla pod nazivom : „HELLO“

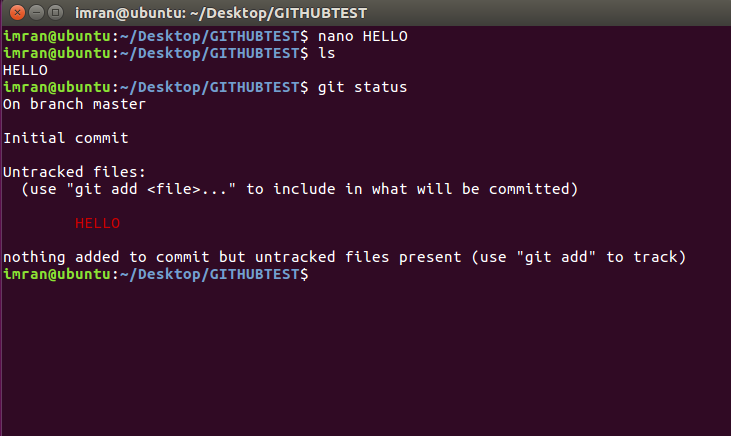
Započinjemo sa kreiranjem foldera „GITHUBTEST“ i nakon ulaska u taj folder moramo inicijalizirati git sa komadnom *git init.*



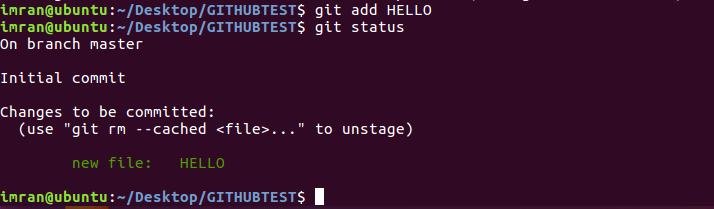
Nakon što smo stvorili git kompaktibilan folder moramo unijeti svoje podatke (email i password) sa Githuba. Komanda *git config --list* pokazuje informacije o korisniku.



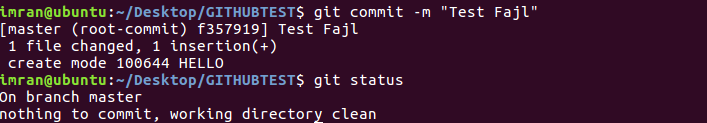
Nakon obavljene registracije možemo pristupiti kreiranju tekstualnog fajla naziva „HELLO“ i provjere njegovog statusa u gitu.



Kao što vidimo tekstualni fajl „HELLO“ nije dodan u INDEX i time nije „tracked“. Da bi smo to uradili moramo izvršiti komandu git add HELLO.



Nakon što je fajl spremljen u INDEX naredni korak je spremanje u HEAD što radimo sa commitom.



I kao konačni korak prebacivanja tekstualnog dokumenta na repozitorij na githubu potrebno je izvršiti „push“ na server.

