

Git - distribuirani sistem za kontrolu verzije

Git je trenutno najpopularniji i najrasprostranjeniji sistem za kontrolu verzije, namenjen praćenju promena izvornog koda tokom procesa razvoja softvera. Iako je razvijen sa idejom podrške koordinaciji tima programera koji zajedno radi na određenom projektu, može biti korišćen i u generalne svrhe, za praćenje promena u okviru proizvoljne kolekcije datoteka. Kako se projekat razvija i evoluira, članovi tima obavljaju testiranje, otklanjaju greške i dodaju nove delove koda znajući da bilo koja verzija koda može biti rekonstruisana u proizvoljnom trenutku. Osobe koje učestvuju u razvoju mogu na osnovu uvida u projektnu istoriju da saznaju:

- Koje su promene unete?
- Ko ih je uneo?
- Kada su promene nastale?
- Zašto su promene bile neophodne?

Git je primer **distribuiranog sistema** za kontrolu verzije koji omogućava pun pristup svakoj datoteci, grani i iteraciji razvoja i svakom korisniku dozvoljava pristup potpunoj i zaokruženoj istoriji svih promena. Za razliku od centralizovanih sistema za kontrolu verzije, distribuiranim sistemima kao što je Git nije neopodna permanentna veza sa centralnim repozitorijumom.

Repozitorijum, odnosno Git projekat predstavlja kolekciju datoteka i direktorijuma u vezi sa projektom, uključujući i istoriju revizija za svaku pojedinačnu datoteku. Istorija predstavlja niz verzija iste datoteke u različitim vremenskim trenucima kada je izvršena **operacija postavljanja (eng. commit)**. Svaka novoizvršena operacija postavljanja nadovezuje se na prethodnu, tako da se u logičkom smislu dobija povezana lista koja može biti razgranata tako da prati različite grane razvoja istog projekta. Svaki učesnik koji poseduje kopiju repozitorijuma ima na raspolaganju kompletnu bazu koda koji pripada projektu, zajedno sa istorijom njegovog razvoja. Korišćenjem osnovnih alata u komandnoj liniji, ili nekog od grafičkih okruženja (GUI), korisnicima su omogućene osnovne operacije interakcije sa istorijom, kloniranja repozitorijuma, postavljanja novih verzija koda, kreiranja i spajanja grana, poređenja različitih verzija koda i sl.

Rad sa repozitorijumima olakšava organizovanje i zaštitu koda, bez čega bi rad na velikim *open-source* projektima bio praktično nezamisliv. Učesnici u razvoju mogu ispravljati greške ili dodavati nove funkcionalnosti bez bojazni da će time pokvariti ili poremetiti glavni tok razvoja projekta. Ovo je omogućeno primenom tematskih grana koje se mogu na jednostavan način kreirati, pripojiti glavnom toku ili proglasiti zastarelim, po potrebi. Uz pomoć mrežnih platformi kao što su **GitHub** ili **Bitbucket**, otvaraju se dodatne mogućnosti kada je u pitanju saradnja i transparentnost tokom razvoja projekta. Javni repozitorijumi omogućavaju razvojnim timovima da sarađuju i doprinose zajedničkom cilju stvaranja najboljeg mogućeg proizvoda.

Osnovne Git komande

Odličan izvor informacija u vezi sa osnovnim i naprednim funkcionalnostima Git-a dostupan je u formi elektronske knjige:

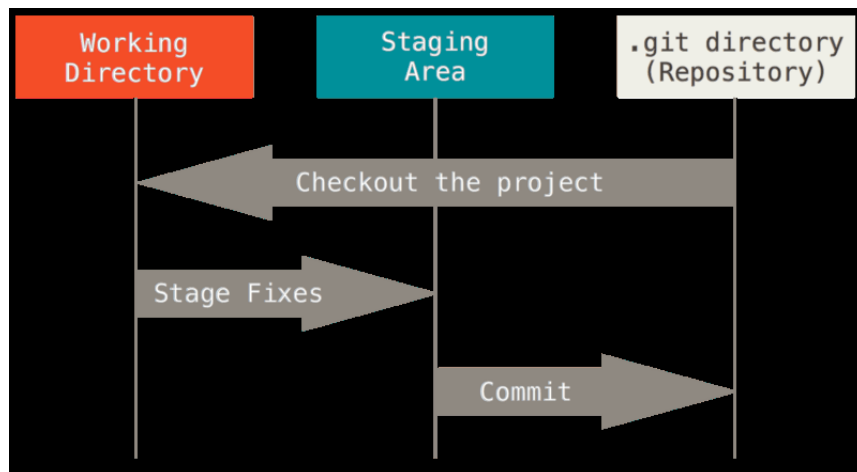
<https://git-scm.com/book/en/v2>

Nakon instalacije odgovarajuće verzije Git-a (za *Linux*, *Windows* ili *macOS*), operacije je moguće obavljati putem komandi koje se izvršavaju iz komandne linije.

Datoteke koje pripadaju projektu u svakom trenutku se mogu nalaziti u jednom od tri osnovna stanja:

- Postavljeno (**committed**) – podrazumeva da je datoteka bezbedno smeštena u lokalnu bazu podataka
- Izmenjeno (**modified**) – sadržaj datoteke je izmenjen od poslednjeg postavljanja, ali sadržaj još nije pohranjen u bazu
- Spremno za postavljanje (**staged**) – izmenjena datoteka je označena za postavljanje

U skladu sa tim, razlikujemo tri glavne sekcije Git projekta: Git direktorijum (**.git directory**), radni direktorijum (**working directory**) i pripremna zona (**staging area**).



Git direktorijum je mesto gde se čuvaju metapodaci i baza podataka u vezi sa projektom. Ovo je najvažniji deo Git sistema i tu se smeštaju podaci prilikom kloniranja repozitorijuma, kao i prilikom postavljanja novih verzija.

Radni direktorijum sadrži datoteke koje predstavljaju stanje tekuće verzije projekta. Tekuća verzija se generiše na osnovu podataka iz baze i na raspolaganju je korisniku za unošenje izmena.

Pripremna zona je datoteka, koja je obično smeštena u Git direktorijumu i koja sadrži informacije o tome šta tačno treba da sadrži sledeća verzija koja će biti generisana prilikom postavljanja (*commit*). U Git terminologiji ova zona se naziva još i „*index*“.

Tok rada se obično odvija u tri faze koje se sukcesivno smenjuju:

1. Korisnik modifikuje sadržaj datoteka u radnom direktorijumu
2. Korisnik selektivno označava one datoteke koje trebaju biti postavljene u repozitorijum, čime se one prebacuju u pripremnu zonu
3. Obavlja se operacija postavljanja (*commit*), čime se preuzimaju datoteke iz pripremne zone i permanentno se smeštaju u Git repozitorijum

git init

Komanda **git init** Inicijalizuje novi Git repozitorijum u tekućem direktorijumu i počinje sa njegovim praćenjem. Na ovaj način se dodaje skriveni direktorijum (.git direktorijum) u kojem je smeštena interna baza podataka neophodna za praćenje verzije:

```
git init
```

git clone

Komanda **git clone** kreira lokalnu kopiju postojećeg repozitorijuma sa servera. Klonirani repozitorijum sadrži sve izvorne datoteke, zajedno sa istorijom promena i granama razvoja. Primer:

```
git clone https://github.com/me/repo.git
```

git status

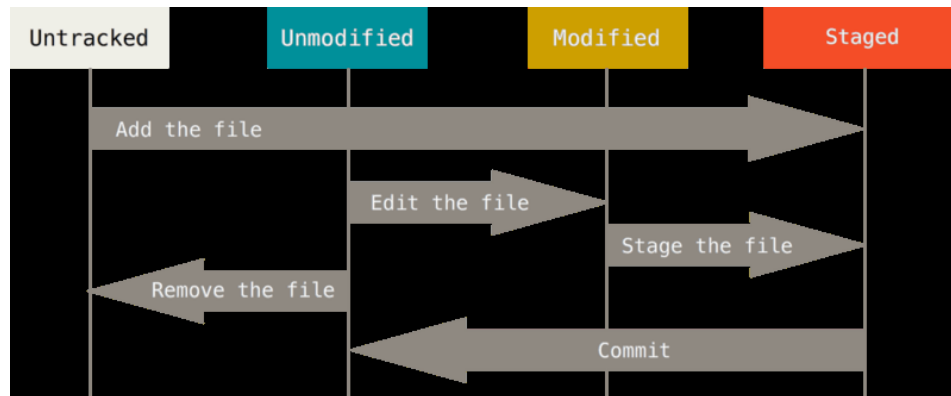
Svaka datoteka u radnom direktorijumu može se nalaziti u jednom od dva stanja:

- Nepraćena datoteka (**untracked**)
- Praćena datoteka (**tracked**)

Praćena je ona datoteka o čijim eventualnim promenama Git sistem vodi računa. Takva datoteka može se nalaziti u jednom od tri stanja:

- Neizmenjena (**unmodified**) – datoteka čiji sadržaj nije izmenjen od poslednjeg postavljanja
- Izmenjena (**modified**) – datoteka čiji sadržaj je promenjen
- Pripremljena za postavljanje (**staged**) – prilikom sledećeg postavljanja, biće postavljene samo one datoteke koje se nalaze u ovom stanju, tj. koje se nalaze u pripreмноj zoni (*staging area*)

Promene stanja datoteka i operacije koje dovode do njih prikazane su na slici:



Komanda **git status** služi za proveru stanja datoteka u radnom direktorijumu. Primer:

```
$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
nothing to commit, working directory clean
```

git add

Komanda **git add** označava izmenjene datoteke za postavljanje, čime ih prebacuje u pripremnju zonu. Ovo je prvi korak u kreiranju nove verzije koda. Sve promene koje su prebačene u pripremnju zonu će prilikom postavljanja postati integralni deo sledeće verzije, a samim tim i projektne istorije.

git commit

Komanda **git commit** snima promene iz pripremnju zone u repozitorijum, čime se kompletira proces kreiranja nove verzije. Novostvorena verzija predstavlja presek stanja projekta u određenom vremenskom trenutku. Sve što je prethodno prebačeno u pripremnju zonu pomoću komande **git add**, postaće deo nove verzije kreirane komandom **git commit**.

git branch

Komanda **git branch** kreira novu granu razvoja koda.

git merge

Komanda **git merge** spaja promene u kodu nastale u dvema odvojenim granama razvoja koda.

git pull

Komanda **git pull** ažurira sadržaj lokalnog repozitorijuma sadržajem sa udaljenog servera.

git push

Komanda **git push** ažurira repozitorijum na udaljenom serveru promenama napravljenim u lokalnu.

Zadatak:

Kreirati korisnički nalog na GitHub-u. Zatim kreirati udaljeni repozitorijum, klonirati ga u lokalu i demonstrirati rad osnovnih Git komandi iz komandne linije.