

Git za gike

Dokument opiše delovanje sistema za nadzor različic [Git](#) za nekoga, ki želi vedeti, kako stvari delujejo.

Kaj je Git?

[Git](#) je sistem za upravljanje različic. Kaj to pomeni?

Git hrani vsebino direktorija z datotekami in celotno zgodovino sprememb. Zgodovina sprememb se hrani v obliki posnetkov celotne vsebine v določenih trenutkih.

Za neučakane

Samostalniki:

- **Vnos** (angl. **commit**) je posnetek trenutnega stanja projekta, shranjen kot vozlišče v zgodovinskem grafu (DAG), ki vsebuje spremembe datotek ter metapodatke (avtor, čas, sporočilo).
- **Zgoščena vrednost vnosa** (angl. **commit hash**) je 40-mestna heksadecimalna vrednost, izračunana s SHA-1, ki enolično identificira vnos na podlagi njegove vsebine.
- **Veja** (angl. **branch**) je premična oznaka, ki kaže na določen vnos v zgodovini in se samodejno premakne naprej, ko dodajamo nove vnose. Veje omogočajo vzporedne razvojne linije z različnimi spremembami.
- **Oznaka** (angl. **tag**) je statična oznaka, ki trajno kaže na določen vnos. Za razliko od veje se oznaka, nikoli ne premika samodejno, zato se uporablja predvsem za označevanje pomembnih točk v zgodovini, kot so izdaje ali stabilne verzije.
- **Delovna kopija** (angl. **workout copy**) je direktorij v katerem urejamo datoteke, ki jih nato vnesemo v Git. V delovni kopiji imajo na začetku datoteke isto vsebino kot je vsebina trenutno aktivnega vnosa (HEAD). Spremembe, ki jih nauredimo na delovni kopiji lahko zabeležimo v nov vnos.
- **Oddaljen repozitorij** (angl. **remote**) je povezava(url) na oddaljen repozitorij, s katerim izmenjujemo vsebino.

Glagoli (akcije):

- **Checkout** prenese vsebino vnosa v delovno kopijo: `git checkout neka-veja`
- **Commit** ustvari nov vnos: `git commit -m 'Sporočilo'`
- **Add** doda vsebino, ki bo v naslednjem vnosu: `git add dodaj_me.txt`
- **Pull** pobere vsebino iz oddaljenega repozitorija in uskladi lokalno vejo z oddaljeno: `git pull`
- **Push** potisni lokalne vnose na oddaljeni repozitorij in uskladi oddaljeno vejo z lokalno: `git push`
- **Fetch** pobere nove vnose, veje in oznake iz oddaljenega repozitorija: `git fetch`
- **Reset** spremeni kam kaže trenutno izbrana veja: `git reset a239f9e91`
- **Merge** ustvari nov vnos, ki združi dve ločeni veji v eno: `git merge main`
- **Rebase** prestavi vnose v trenutno izbrani veji na izbran vnos: `git rebase main`

Vnos: posnetek stanja

Osnovna enota v Gitu je **Vnos** (angl. **commit**). Vnos je posnetek stanja zabeleženih datotek v trenutku, ko je bil ustvarjen. Poleg vsebine datotek vsak vnos vsebuje še metapodatke o avtorju, datumu vnosa in opisom sprememb. Vsakemu vnosu je prirejena **zgoščena vrednost vnosa**, ki je 40-mestna heksadecimalna vrednost, izračunana s SHA-1, in je natanko določena z vsebino shranjenih datotek in metapodatkov vnosa. Vnose in vsebino datotek hrani Git v [skladišču objektov](#). Do objektov v skladišču lahko dostopamo, če poznamo njihovo *zgoščeno vrednost*. Objekti, ki jih Git hrani v skladišču so **vnosi**, **posnetki direktorijev** in **posnetki posameznih datotek**.

zgoščena vrednost: 8dd6d4bdaeff93016bd49474b54a911131759648
tree 65c47feec7465e80492620a48206793e078702e0
parent 16f2994757f1213935b8edb9ae7fee3a8e9ec98d
author MV <mv@example.com> 1765235698 +0100
committer MV <mv@example.com> 1765235698 +0100
Dodaj bla

Tabela 1: Primer vnosa v Gitu. Vnos vsebuje zgoščeno vrednost posnetka direktorija(tree), zgoščeno vrednost starševskega vnosa (parent) in metapodatke. Tudi sam vnos je natančno določen z zgoščeno vrednostjo.

Posnetki direktorijev so v Gitu posebne vrste objekti tipa tree. Vsebujejo zgoščene vrednosti in metapodatke o datotekah in direktorijih, ki jih vsebuje.

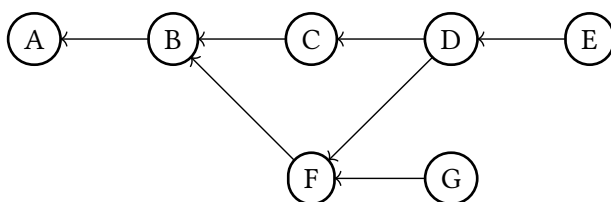
zgoščena vrednost: d934342ca420dd0d9828782c7103103f2922d2a6			
100644	blob	76018072e09c5d31c8c6e3113b8aa0fe625195ca	bar.txt
100644	blob	ba0e162e1c47469e3fe4b393a8bf8c569f302116	foo.txt
040000	tree	3b8bfca88b2cc4127ce5909eb3a7395e8b5f2b6a	podmapa

Tabela 2: Primer posnetka direktorija v Gitu (objekt tipa tree). Posnetek vsebuje zgoščene vrednosti datotek in direktorija, ki jih vsebuje. Uporaba zgoščenih vrednosti natančno določa vsebino posnetka direktorija.

Skladišča objektov v Gitu je **skladišče vsebinsko naslovljivih objektov**. Dostop do objekta je mogoč, če poznamo **zgoščeno vrednost** njegove vsebine. To pomeni, da je referenca na posamezen objekt v Gitu preprosto zgoščena vrednost(angl. hash) vsebine tega objekta. Po drugi strani je vsebina objekta določena z njegovo zgoščeno vrednostjo. To pomeni, da lahko enostavno preverimo verodostojnost vsebine, ki je shranjena v Gitu.

Zgodovinski graf sprememb

Posamezni vnosi so povezani v **usmerjen acikličen graf (DAG)**, ki predstavlja zgodovino sprememb. Vsak **vnos** je **vozlišče** v grafu. Vsak vnos izhaja iz enega ali več starševskih vnosov. Izjema je prvi vnos. **Povezave** v grafu povezujejo vnose z njihovimi starši.



Slika 1: Vnosi v Gitu kot usmerjen graf. Vsak vnos(razen prvega) ima povezavo na vnose iz katerih izhaja.

Reference: veje in značke

V delu!

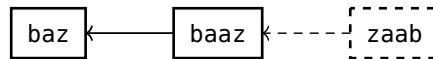
Ukazi v git

Za konec so še kratke ilustracije kaj v Gitu počnejo posamezni ukazi.

Git add

Zabeleži spremembe, ki bodo vključene v naslednji vnos:

```
git add foo.bar
```



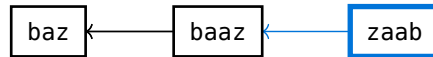
Slika 2: Zabeleži, katere spremembe bodo v naslednjem vnosu.

Ustvari **čakalnico** za spremembe, ki bodo vpisane naslednjem vnosu (angl. **staging area**).

Git commit

Spremembe zapiši v nov **vnos (commit)**:

```
git commit -m "Spremeni baaz v zaab!"
```

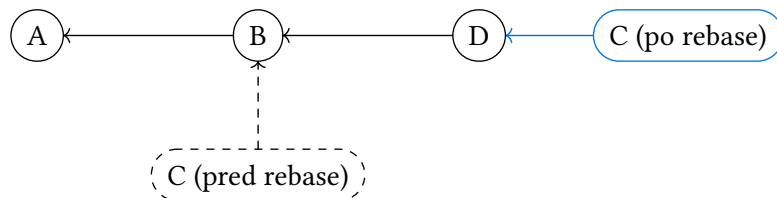


Slika 3: Zapiši spremembe v nov vnos

Ustvari novo vozlišče v grafu.

Git Rebase

Premakne zaporedje vnosov na novo osnovo.



Slika 4: Rebase: commit C se premakne, da temelji na D.

Git Reset

Premakne oznako veje na drug commit.

