

# Git – rad sa udaljenim repozitorijumom

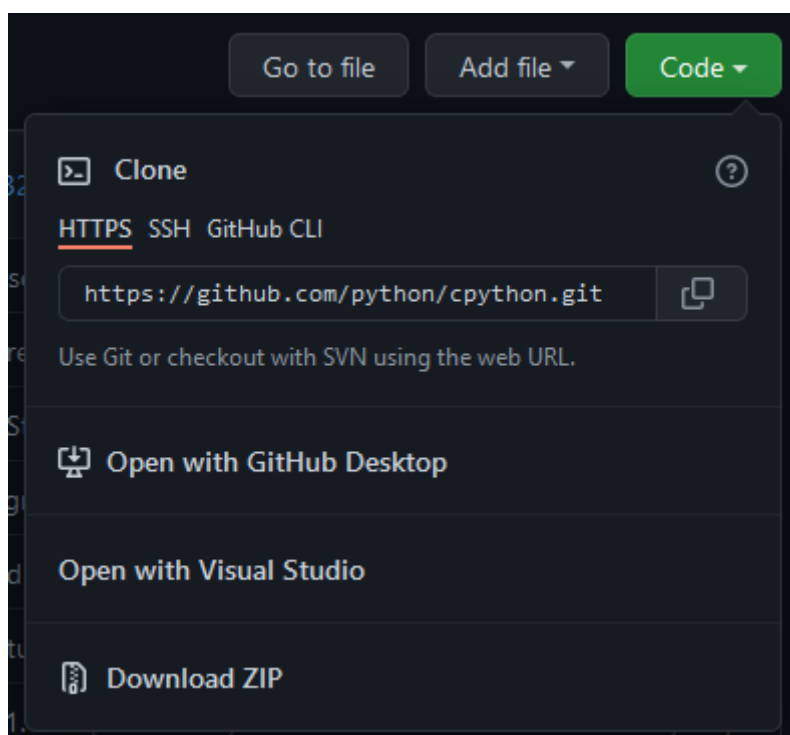
## Udaljeni repozitorijumi i udaljene grane

Naučio si kako da radiš sa gitom u lokalu, a sad je vrijeme da naučimo kako možemo da razmjenjujemo izmjene sa kolegama. Pored lokalnih repozitorijuma postoje i udaljeni repozitorijumi. Udaljeni repozitorijum je repozitorijum koji se nalazi na nekom drugom računaru ili serveru.

Udaljene repozitorijume ćemo praviti na nekoj od hosting platformi, kao što su *GitHub*/*GitLab*. *GitHub* repozitorijum je repozitorijum isti kao i tvoj lokalni, ali se nalazi na nekom *GitHub*-ovom serveru. Ove udaljene repozitorijume možeš da podešavaš tako da ih prilagodiš svojim specifičnim potrebama. Na *GitHub*-u napravi novi prazan repozitorijum.

Da bi mogli da razmjenjujemo izmjene između lokalnog i udaljenog repozitorijuma, potrebno je da izvršimo određena podešavanja. Podešavanja uključuju dodavanje udaljenih repozitorijuma našem lokalnom repozitorijumu. Nakon dodavanja možemo slati i dobavljati izmjene sa udaljenog repozitorijuma.

Link do udaljenog repozitorijuma možeš vidjeti tako što ćeš pritisnuti na zeleno „Code“ dugme na *GitHub* stranici prikazano na slici ispod.



Možeš da odabereš način na koji će se vršiti autentifikacija prilikom razmjene izmjena između repozitorijuma: preko HTTPS protokola ili putem SSH. Savijet je da koristiš SSH ukoliko radiš na svom računaru. Ukoliko radiš na računaru iz učionice, koristi HTTPS i kreiraj Personal Access Token koji ćeš koristiti umjesto šifre. Kopiraj putanju do udaljenog repozitorijuma (uptanja je oblika: [git@github.com:python/cpython.git](https://github.com/python/cpython.git) ili <https://github.com/python/cpython.git>). Uputstvo za kreiranje podešavanje SSH je na ovom [tutorijalu](#), a Personal Access Token [ovdje](#).

Obično bude jedan udaljeni repozitorijum koji se zove `origin`. Ovaj udaljeni repozitorijum koristimo kao „single source of truth“ odakle dobavljamo i gdje šaljemo izmjene. Da bi podesili udaljeni repozitorijum za vaš lokalni repozitorijum izvrši naredbu: `git remote add origin <url do udaljenog repozitorijuma>`. Nakon ove komande `origin` možemo koristiti umjesto linka do našeg repozitorijuma (`origin` možeš posmatrati kao promjenljivu u koju smo smjestili link do repozitorijuma, kad nam u budućnosti bude trebalo da kažemo na koji repozitorijum šaljemo ili dobavljamo izmjene, koristićemo `origin` umjesto pune putanje). Praksa je da se glavni udaljeni repozitorijum zove `origin`. Za link do udaljenog repozitorijuma stavite link do *GitLab/GitHub* repozitorijuma koji ste napravili ranije.

Da bi provjerili koji udaljeni repozitorijumi su dodati izvršite: `git remote -v`. Prodiskutuj sa asistentom rezultat izvršavanja ove naredbe.

Pored lokalnih grana postoje i udaljene grane. Udaljene grane se nalaze na udaljenim repozitorijumima. Više o granama u nastavku.

## Slanje izmjena

Kad si podesio udaljeni repozitorijum, vrijeme je da na njega pošalješ sve što si razvio u lokalu. Da bi to odradio iskoristi komandu `push`. Ova komanda šalje sve izmjene sa lokalne grane na udaljenu granu. Šalju se samo nove izmjene koje ne postoje na udaljenoj grani. **Da bi izvršili komandu potrebno je da kažemo gitu na koji udaljeni repozitorijum šaljemo izmjene i na koju granu tog repozitorijuma.** Pošalji izmjene na udaljeni repozitorijum koji si ranije kreirao sa: `git push origin <ime udaljene grane>`. (dakle ovdje šaljemo izmjene koje se nalaze na grani na kojoj smo trenutno na udaljeni repozitorijum `origin` na udaljenu granu koju navedemo). Obično ime udaljene grane odgovara imenu lokalne grane na kojoj se trenutno nalazimo, mada to ne mora da bude slučaj. Ukoliko grana koju smo naveli ne postoji na udaljenom repozitorijumu, nova grana sa tim imenom će biti kreirana.

Da ne bi svaki put kucali ime udaljenog repozitorijuma i udaljene grane možeš podesiti podrazumijevanu udaljenu granu na koju će se slati izmjene sa tvoje lokalne grane. To radiš sa `git push -u origin <ime udaljene grane>`. Kad ovo podesiš, svaki idući put možeš da šalješ izmjene samo sa `git push`. Ovo znači da smo povezali lokalnu granu sa udaljenom i da će se prilikom komande `git push` izmjene uvijek slati na granu koju smo naveli.

## Dobavljanje izmjena

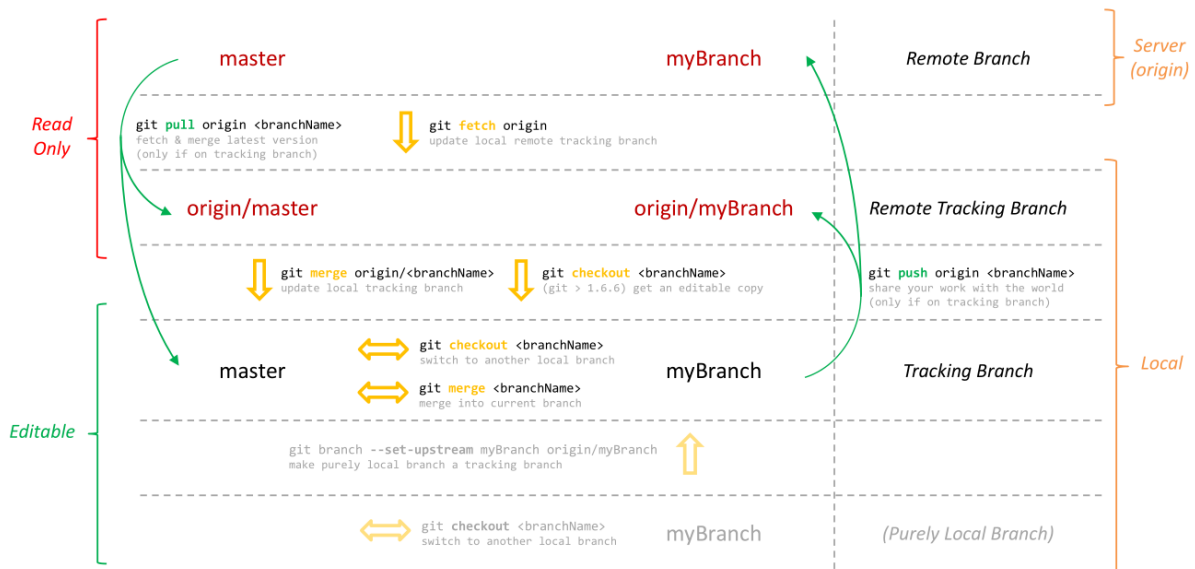
Da bi dobavili izmjene sa udaljenog repozitorijuma koristimo naredbu `git fetch origin`. Ova naredba dovuče sve nove grane i izmjene na svim granama sa udaljenog repozitorijuma. Te izmjene su smještene u takozvanim *remote tracking* grane (grane koje prate udaljene grane). *Remote tracking* grane su oblika `origin/<ime grane>` i one su *read-only* (samo su za čitanje, ne mogu se slati izmjene na njih). Na *remote tracking* grane se dovuku izmjene sa udaljenih grana. Da bi te izmjene integrisali sa našim lokalnim granama radimo `git merge origin/<ime grane>`, isto kao kada spajamo dvije lokalne grane. Kako konflikti nastaju prilikom spajanja lokalnih grana, isto tako mogu nastati prilikom dobavljanja izmjena sa udaljene grane.

Kako se uvijek radi `git fetch origin`, pa `git merge origin/<ime grane>`, git ima skraćenicu za to: `git pull origin <ime grane>`. Savjet je da na početku rada koristiš *fetch/merge* kombinaciju, dok se ne navikneš na tok rada, pa da vremenom pređeš na pull zbog brzine

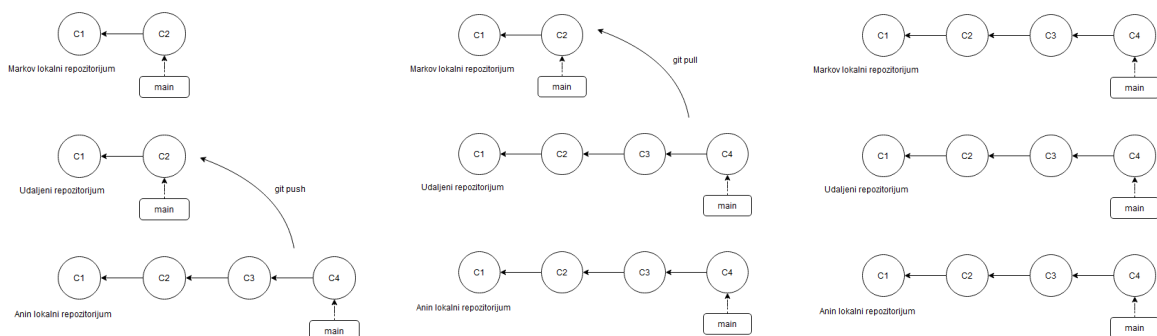
rada. Slično kao i kod push komande, ako se podesi default-na push/pull grana, naredba se može pozivati samo sa `git pull`.

Na slici ispod su prikazani tipovi grana u git-u i komande koje koristimo da upravljamo njima. Detaljno analiziraj sliku sa asistentom i prodiskutuj sve vrste grana.

## Git branches for the rest of us



Na slikama ispod je prikazana razmjena izmjena. Miloš je napravio neke izmjene koje šalje na udaljeni repozitorijum. Kad su izmjene poslate na udaljeni repozitorijum, Marko može da ih preuzme u svoju lokalnu kopiju main-a i nastavi sa razvojem.



## Kloniranje udaljenog repozitorijuma

Postavlja se pitanje kako možemo da preuzmemo stanje udaljenog repozitorijuma kojeg nemamo lokalno na računaru. To radimo sa naredbom `clone` na idući način: `git clone <url do udaljenog repozitorijuma>`. Sa ovim će git preuzeti udaljeni repozitorijum, napraviti novi folder sa njegovim imenom i u njega će smjestiti sve fajlove i sve izmjene zabilježene u istoriji tog udaljenog repozitorijuma. Nakon clone-a možeš nastaviti rad na repozitorijumu. Isprobaj ovu naredbu tako što ćeš klonirati idući repozitorijum: <https://github.com/SIIT-USI/git-clone-test>.

Isprobaj clone naredbu još jednom tako što ćeš klonirati repozitorijum koji je kreirao kolega iz klupe za izradu kalkulator zadatka.

Otvori kolegin projekat i napravi novu granu. Na toj grani napravi fajl pod nazivom README.md u koji kratko opiši šta program radi. Pošalji novu granu na kolegin udaljeni repozitorijum. (Ukoliko nisi u mogućnosti da radiš u paru, možeš da kloniraš svoj kalkulator projekat na neku drugu lokaciju na svom računaru i posmatraš tu kopiju projekta kao kolegin kalkulator).

Nakon što si to odradio na koleginjoj grani, vrati se u svoj kalkulator projekat i u njemu dodaj README.md fajl sa opisom šta program radi i ko ga je napravio. Pošalji sve izmjene na udaljeni repozitorijum.

## Vježbanje rada sa granama

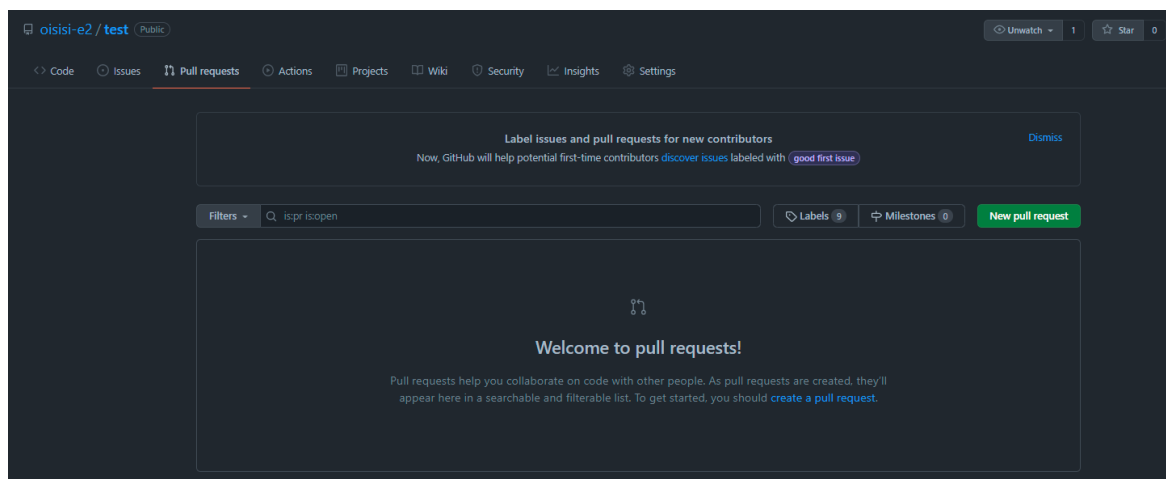
Da bi izvježbao rad sa granama, kako lokalnim, tako i udaljenim, pređi vježbe sa ovog sajta <https://learngitbranching.js.org/> vezane za rad sa granama. Za sad možeš preskočiti vježbe vezane za rebase i interaktivni rebase. Ukoliko si zainteresovan za rebase komande, popričaj sa svojim asistentom.

## Pull request

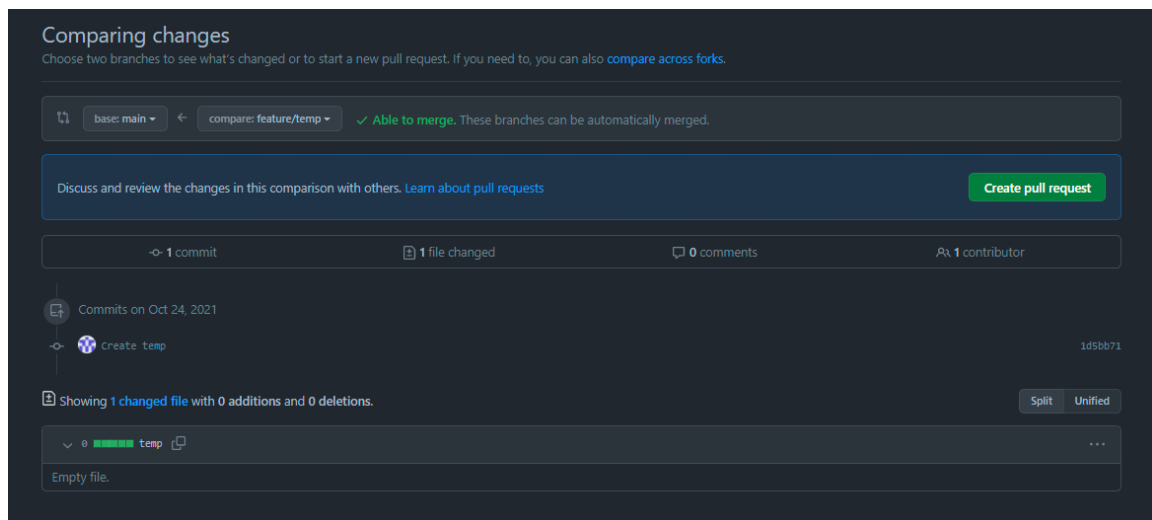
Pull request (PR) je proces putem kog se novi kod integriše na udaljeni repozitorijum. Ideja iza PR je da se kod prije spajanja sa repozitorijumom provjeri za greške, iztestira kako bi se povećao kvalitet koda i projekta. PR je kao neka kontrola merge naredbe, gdje neku granu umjesto da spojiš lokalno, napraviš PR pomoću kog ćeš je spojiti, kolege iz tima pogledaju i provjere tvoje izmjene i spoje ih ako su zadovoljavajuće. Na ovaj način se prije spajanja lako može provjeriti kod za greške. Spajanjem PR-a se i same grane spoje (odradi se merge tvoje grane).

Napravi svoj prvi PR tako što ćeš napraviti PR za granu koju si napravio u kalkulator projektu na udaljenom repozitorijumu tvog kolege. U koracima ispod je opisano kako možeš napraviti novi PR.

PR možeš napraviti putem GitHub-a. Otvori Pull request tab repozitorijuma u kojem želiš da otvoriš PR i pritisnite na “New pull request”.

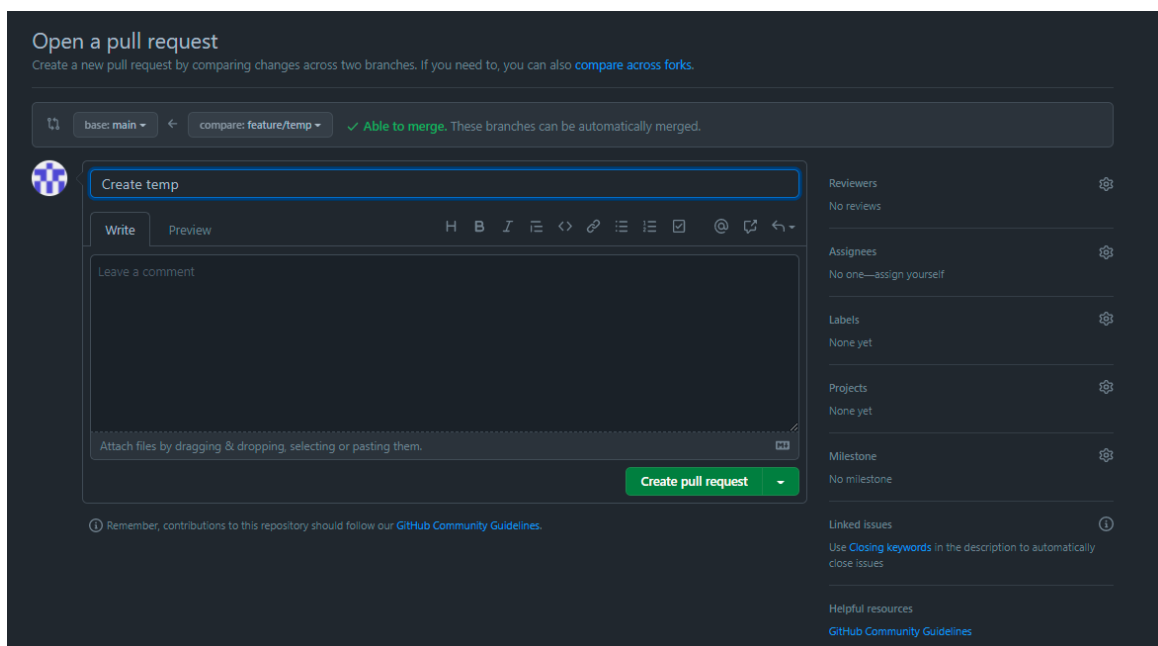


Idući korak je prikazan na slici ispod. Potrebno je da odabereš koju granu želiš da spojiš na koju granu. Ispod možeš da pogledaš pregled commit-ova i pregled izmjena.

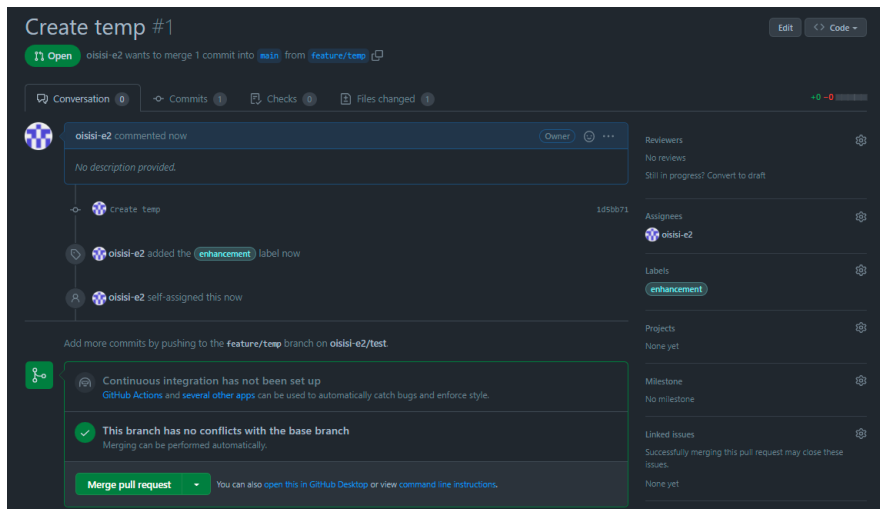


Nakon pritiska na dugme “Create pull request” otvori se prozor sa slike ispod. Tu možemo da podesimo naziv PR-a kao i opis – **šta smo izmijenili i zašto**. Dalje, moguće je podesiti “Reviewers” grupu ljudi koji treba da odrade review novih izmjene. “Assignees” grupa ljudi koja je zadužena za PR – ljudi koju su pravili izmjene. Obično je to osoba koja pravi PR. “Labels” su kao tagovi sa kojima možeš da označiš PR, pa kasnije koristiš za filtriranje.

Kad sve popuniš, pritisni na “Create pull request” da bi napravio PR.

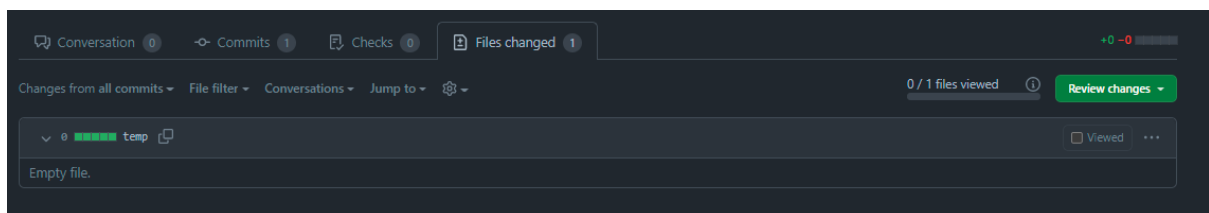


Otvoren PR je prikazan na slici ispod.

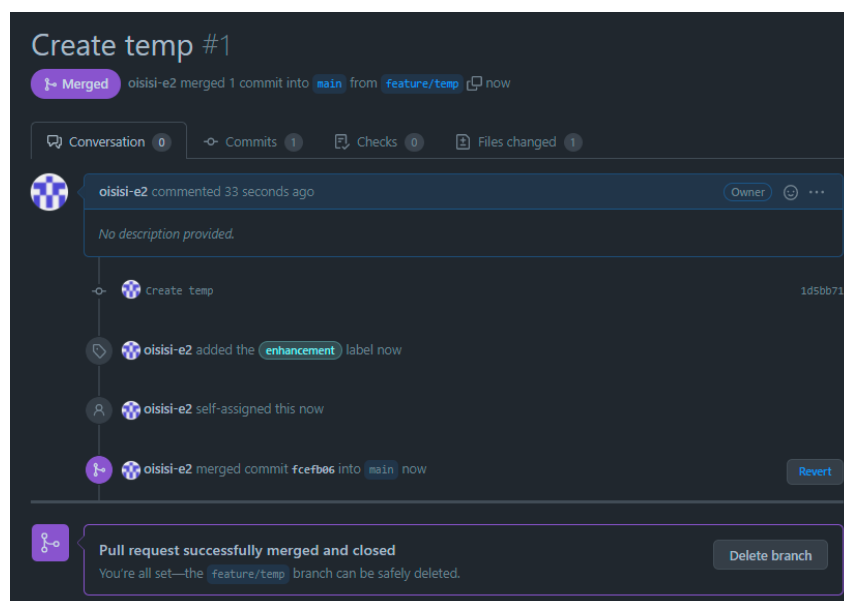


Pod tabom “Files changed” možeš da odradiš review PR-a. Review podrazumijeva da prođeš kroz sve izmjene, testiraš sve, uvjeriš se da radi kako je zamišljeno. Usput ostavljaj komentare na mjestima koje treba izmijeniti ili ako želiš da prodiskutuješ o načinu implementacije sa kolegom koji je napravio PR, i slično. Na dugme „Review changes“ kreiraš review. Možeš da:

- odobriš izmjene i dozvoliš kolegi da spoji izmjene u granu u koju želi
- samo ostaviš komentar bez eksplicitnog odobrenja ili
- možeš da zatražiš izmjene i onemogučiš spajanje dok se ne dodaju tražene izmjene.



Kad dobiješ sve dozvole za merge od svih review-era možeš da odradiš merge i spojiš svoje izmjene. Spojen PR je prikazan ispod.



Zatraži review od svog kolege. Kolega će spojiti tvoj PR nakon što ga odobri.

## Dodatna konfiguracija

Kako bi ubrzali rad sa gitom preporuka je da napravite git aliase. Alias je skraćeno ime za pisanje čestih git komandi. Pogledajte [dokumentaciju](#) za detalje i primjere.

## Literatura i inspiracija

- <https://git-scm.com/book/en/v2> - Pro git knjiga (git Biblija), preporučuje se da pročitaš prva 3 poglavlja
- <https://missing.csail.mit.edu/2020/version-control/> - video i tekstualni materijali MIT-a o git-u
- <http://www.igordejanovic.net/courses/tech/git/> - odlični slajdovi koje je napravio profesor Dejanović, osnovni i napredni pojmovi su detaljno objašnjeni. Dodatno, sve je na srpskom.