

TEHNIČKO VELEUČILIŠTE U ZAGREBU
STRUČNI STUDIJ INFORMATIKE

Salem Velić

ALATI ZA PRAĆENJE RAZVOJA I ODRŽAVANJE
PROGRAMSKOG KODA

Zagreb, svibanj 2016.

Sadržaj

1. Osnovni pojmovi.....	3
2. Uvod u GitHub	4
2.1. Kreiranje repozitorija.....	4
2.2. Dodavanje suradnika.....	6
2.3. Grananje.....	7
2.4. Commit.....	8
2.5. Pull request	9
2.6. Merge	11
3. Zadatak za vježbu	12

1. Osnovni pojmovi

Verzioranje koda: proces zapisivanja promjena u datotekama tijekom vremena tako da se u bilo kojem trenutku možemo vratiti na željeno prethodno stanje. Na primjer, ukoliko testiranjem verzije 1.0 naiđemo na dijelove koda koje je potrebno ispraviti iste ćemo nakon pohrane pohraniti s verzijom 1.1, a s kojom smo zadovoljni. Također, realan je i scenarij da nakon nekog vremena zaključimo da imamo bolju ideju kako modificirati verziju 1.0, te je potrebno „baciti u smeće“ verziju 1.1, vratiti se na 1.0 i od nje napraviti 1.1b. Ovo je samo jedan od mnogih mogućih scenarija kakvi se događaju u programerskom životu.

Git: alat koji omogućuje verzioranje koda. Razvio ga je Linus Torvalds s ciljem lakšeg razvoja Linux kernela zbog čega je vjerojatno u svijetu *open sourcea* stekao auru važnosti. Osim toga, omogućuje i da više osoba sudjeluje istovremeno u jednom projektu. Projekt se može i „otvoriti“ javnosti tako da svatko može dodavati svoje značajke ili ići u drugačijem smjeru te napraviti neki posve novi projekt.

GitHub¹: jedna od najpopularnijih platformi za razmjenu verzioranog koda. Osim nje, još su popularni i Google Code², GitLab³, Bitbucket⁴, Sourceforge⁵ i Microsoftov CodePlex⁶.

¹ <https://github.com>

² <https://code.google.com>

³ <https://about.gitlab.com>

⁴ <https://bitbucket.org>

⁵ <https://sourceforge.net>

⁶ <https://www.codeplex.com>

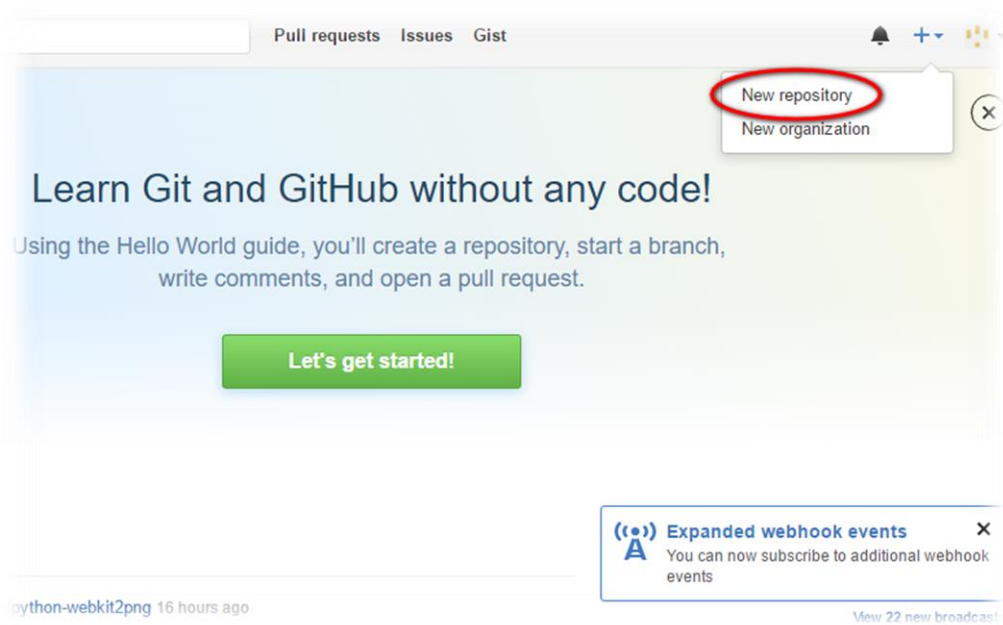
2. Uvod u GitHub

GitHub je hosting servis za Git repozitorije koji omogućuje korištenje svih funkcionalnosti Gita u kombinaciji sa setom drugih mogućnosti specifičnih za GitHub. Ukratko, GitHub je mjesto gdje svaki programer može podijeliti svoj izvorni kod s ostatkom svijeta. Koristi ga preko deset milijuna ljudi koji su zajedno kreirali preko 26 milijuna GitHub repozitorija. U ovoj vježbi će biti prikazano kako napraviti korisnički račun, kreirati repozitorij, načiniti promjene te potom te iste promjene spremiti.

2.1. Kreiranje repozitorija

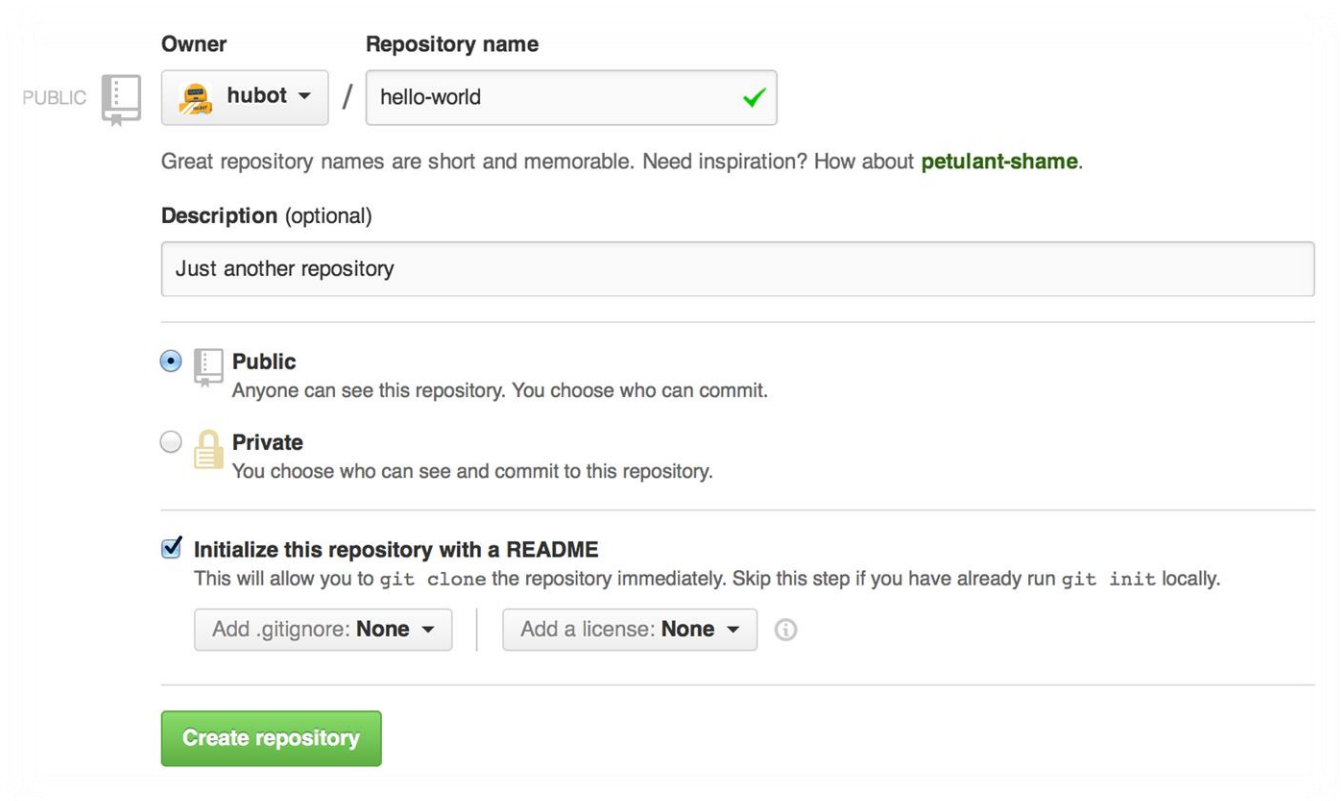
Repozitorij: projekt koji je kreiran i pohranjen u sustavu za verzioniranje koda. Repozitorij sadržava cijelu povijest projekta.

- 1) Registrirajte se na <https://github.com> odabirom korisničkog imena te unosom svoje email adrese i postavljanjem lozinke. Nakon toga kliknite na **Sign up for GitHub**.
- 2) U gornjem desnom kutu kliknite na znak „+“ a zatim odaberite **New repository** kao što je prikazano slikom 1.



Slika 1.

- 3) Nazovite ga hello-world i dodijelite mu kratki opis.
- 4) Također je moguće odabrati hoće li repozitorij biti javan (**Public**) te time dostupan svima, ili privatan (**Private**) te time dostupan samo vama odabranim osobama. Privatni repozitorij nije besplatan.
- 5) Odaberite **Initialize this repository with a README**. Još je moguće dodati **.gitignore**. Gitignore sadrži datoteke za koje ne postoji potreba da se spremaju u povijest projekta. Git će se prema tim datotekama ponašati kao da ne postoje. Osim toga, može se dodati i **license** kojim definiramo autorska prava. Kada odabrati koju licencu možete provjeriti na linku - <http://choosealicense.com>.



The screenshot shows the GitHub repository creation interface. At the top, there's a 'PUBLIC' label with a monitor icon. Below it, the 'Owner' is set to 'hubot' and the 'Repository name' is 'hello-world', which is marked with a green checkmark. A hint text says: 'Great repository names are short and memorable. Need inspiration? How about **petulant-shame**.' The 'Description (optional)' field contains the text 'Just another repository'. Under the 'Visibility' section, the 'Public' radio button is selected, with the text 'Anyone can see this repository. You choose who can commit.' The 'Private' option is also visible with the text 'You choose who can see and commit to this repository.' The 'Initialize this repository with a README' checkbox is checked, with a note: 'This will allow you to `git clone` the repository immediately. Skip this step if you have already run `git init` locally.' Below this, there are two dropdown menus: 'Add .gitignore: None' and 'Add a license: None', followed by an information icon. At the bottom, there is a green 'Create repository' button.

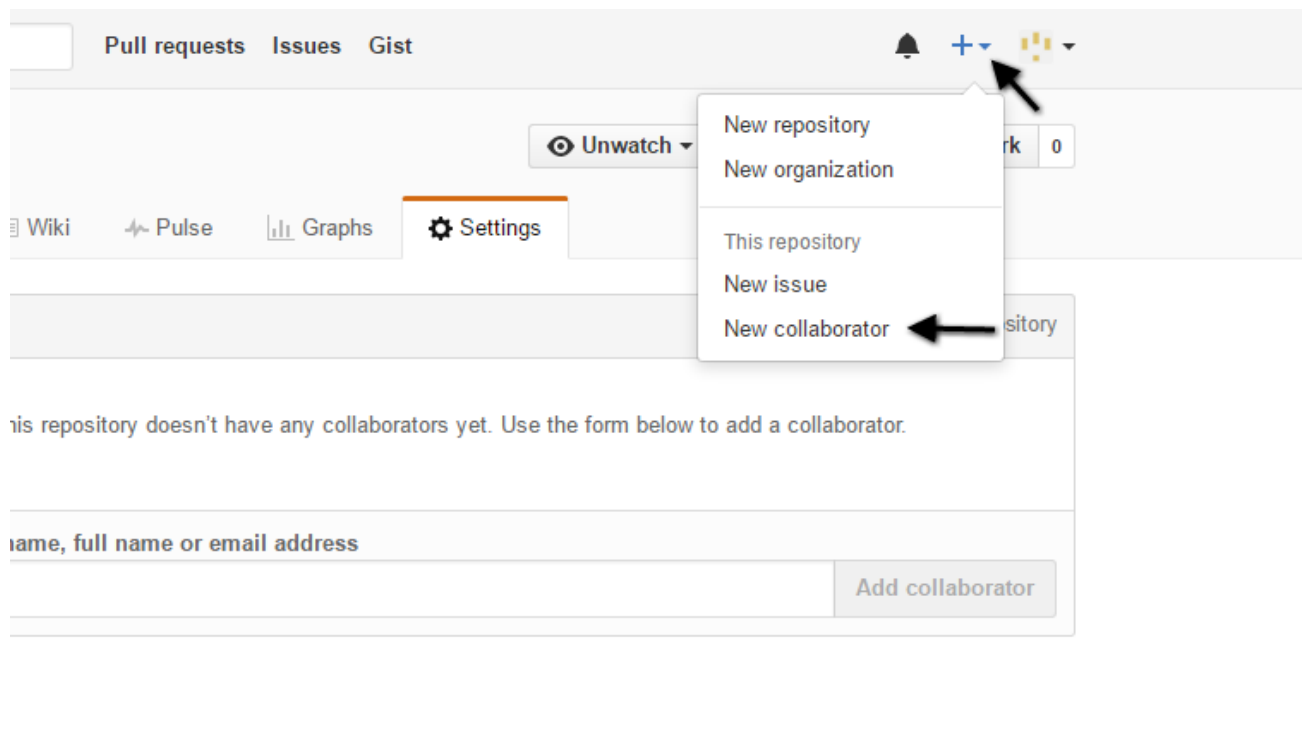
Slika 2

- 6) Kliknite **Create repository**.

2.2. Dodavanje suradnika

Collaborators – osobe koje imaju ovlasti za pisanje i čitanje u repozitoriju.

- 1) Dodajte nove suradnike prema uputama na slici 3. Možete ih pronaći pretragom po imenu, korisničkom imenu ili email adresi, a dodajete ih klikom na **Add collaborator**.

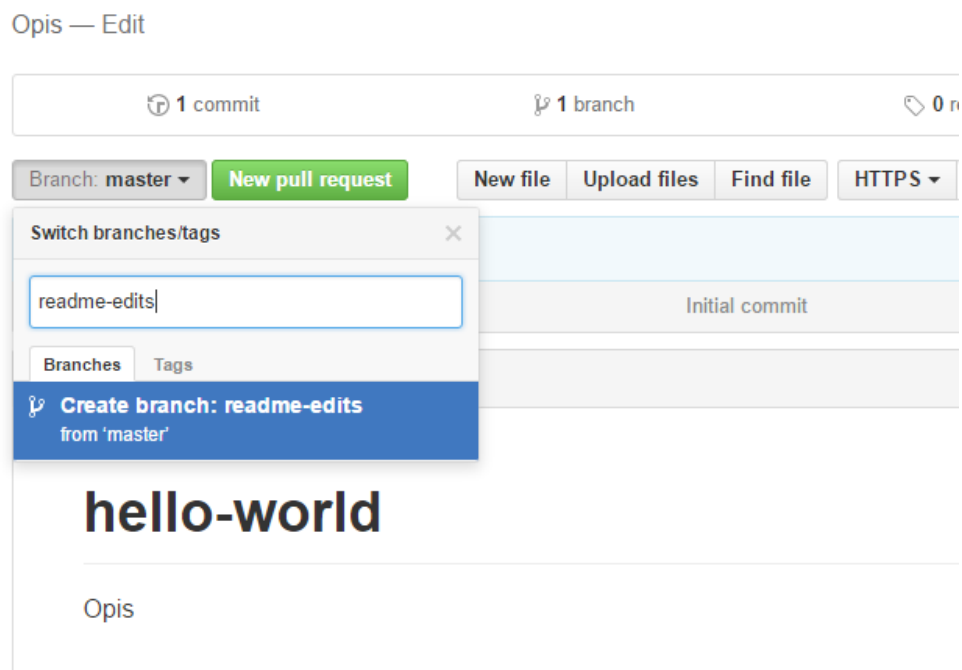


Slika 3

2.3. Grananje

Svaki Git repozitorij dolazi s glavnom granom koja se zove **Master**. No, moguće je napraviti više grana (**branch**) i sve promjene koje se dogode u sporednoj grani neće utjecati na izvorni projekt.

- 1) Kliknite na **Branch: master** na lijevoj strani prozora.
- 2) Nazovite granu „readme-edits“.
- 3) Odaberite **Create branch** ili pritisnite enter.



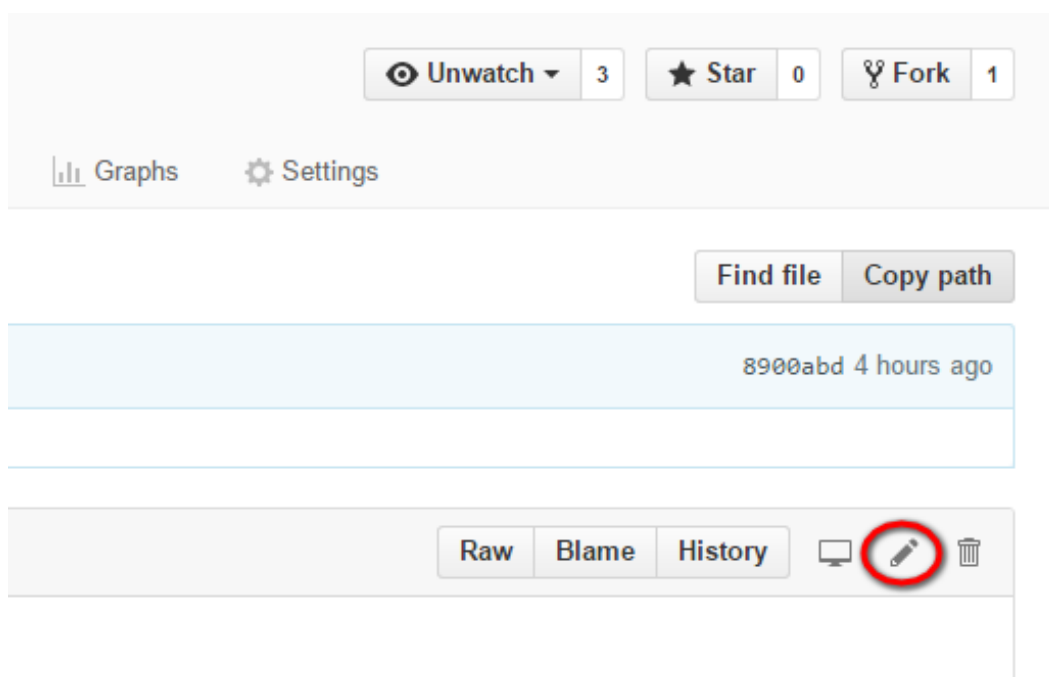
Slika 4

Sada imate dvije grane: **master** i **readme-edits** koje izgledaju identično.

2.4. Commit

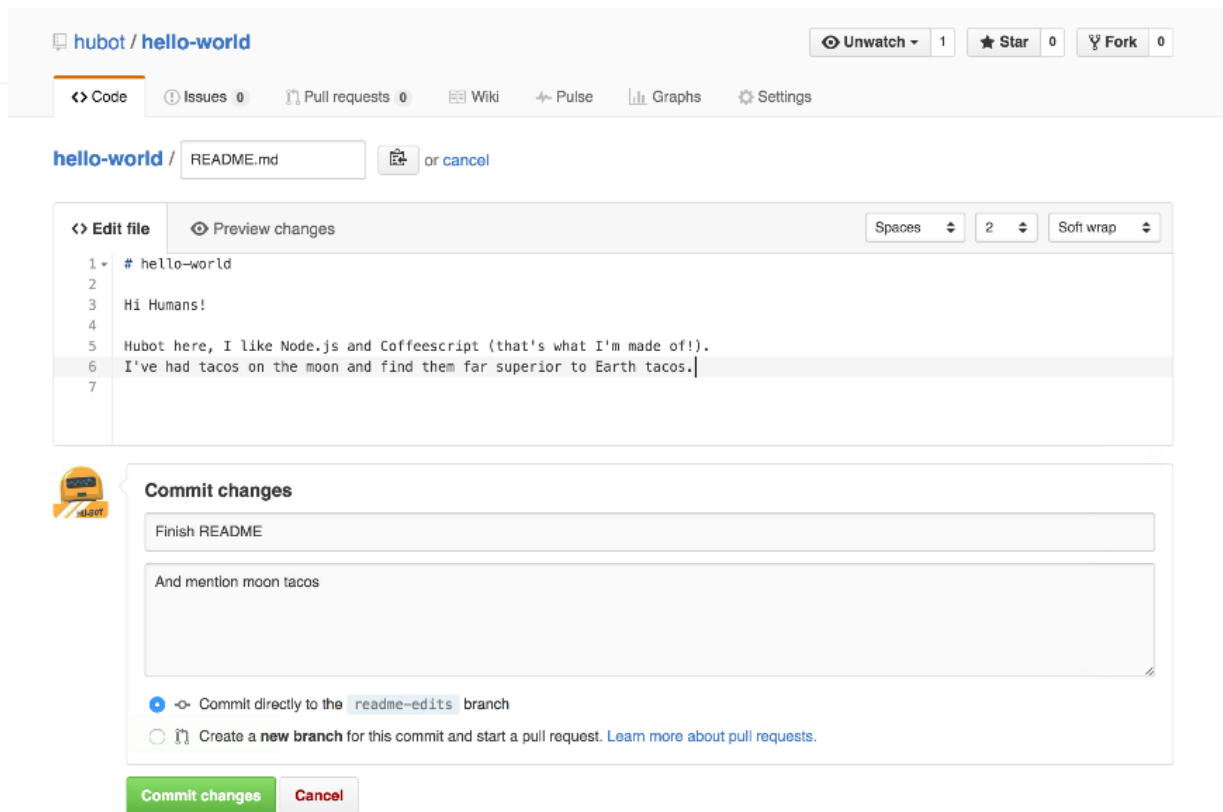
Commit: točka u povijesti projekta koja je promijenila neku značajku u odnosu na zadnju točku.

- 1) Kliknite na datoteku **Readme.md**.
- 2) Zatim na ikonu olovke na desnoj strani koja se vidi na slici 5.



Slika 5

- 3) U editoru, napišite nešto o sebi.
- 4) Opišite kakvu ste promjenu napravili.
- 5) Moguće napraviti commit direktno na master granu ili kreirati novu granu i pokrenuti pull request. O pull requestu će biti govora kasnije.
- 6) Kliknite tipku **commit changes**.



Slika 6

Napravljene promjene će biti vidljive samo u **readme-edits** grani, pa sada ta grana ima drugačiji sadržaj od **master** grane.

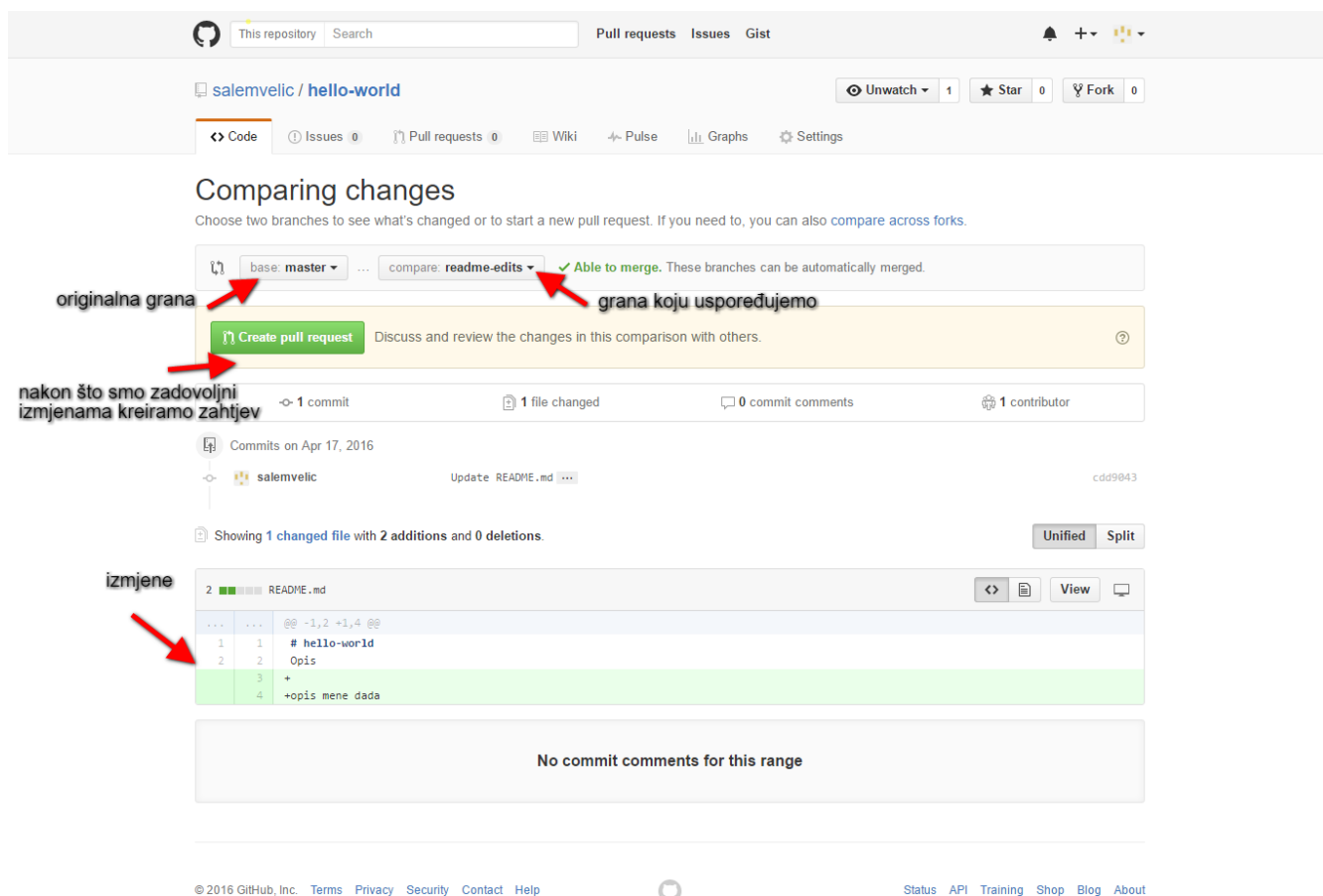
2.5. Pull request

Push – upload promjena na udaljeni repozitorij.

Pull request: zahtjev vlasniku udaljenog repozitorija (na kojeg nemamo ovlasti pushati) da preuzme izmjene koje smo mi napravili.

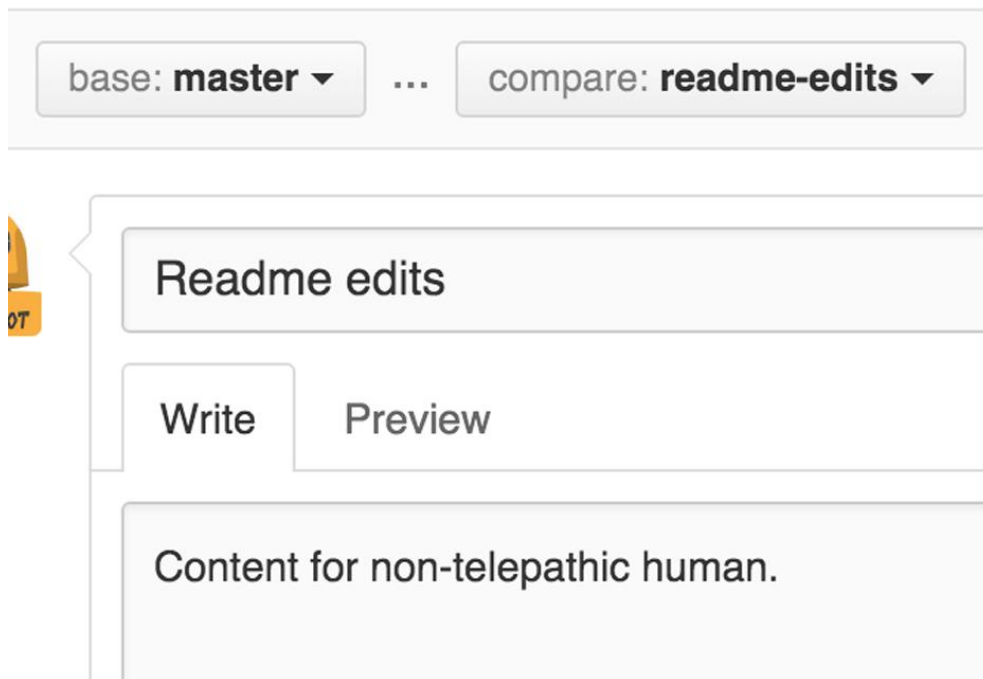
- 1) Kliknite na **Pull requests** te zatim odaberite **New pull request**.
- 2) Odaberite granu koju ste promijenili, **readme-edits**, da bi je usporedili s **master** granom (originalom).

- 3) Provjerite promjene da bi bili sigurni da ste izmijenili upravo ono što ste željeli.
- 4) Kada ste zadovoljni izmjenama, kliknite **Create pull request**.



Slika 7

- 5) Dajte svom zahtjevu naslov i napišite kratki opis promjena.



Slika 8

6) Kada ste završili kliknite **Create pull request**.

2.6. Merge

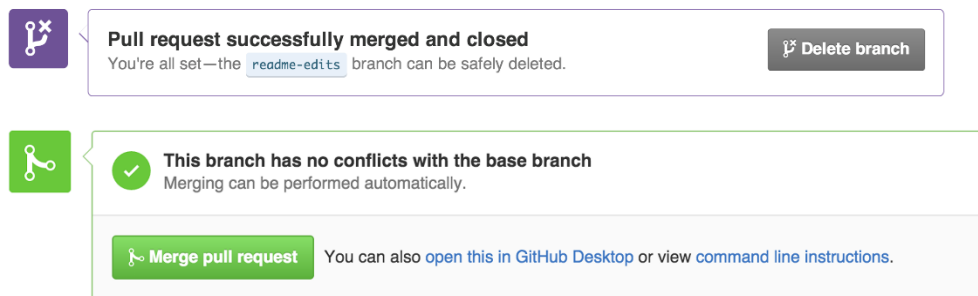
Merge: preuzimanje izmjena iz jedne grane u drugu granu. To nije „spajanje“ jer bi rezultat spajanja bila samo jedna grana dok nakon *mergea* dvije grane nastavljaju svoj život. Jedino što se sve izmjene koje su do tog trenutka rađene u jednoj grani preuzimaju u drugu granu.

Restore: mijenja postojeću povijest projekta tako da se vrati na prijašnju verziju.

Revert: mijenja postojeću povijest projekta tako da doda novi commit koji miče sve izmjene uvedene u čvoru koji želimo promijeniti.

- 1) Kliknite na **Merge pull request** kako bi *mergeali* promjene na **master**.
- 2) Kliknite na **Confirm merge**.

- 3) Nakon što su promjene uključene, obrišite granu klikom na **Delete branch** u ljubičastoj kocki.



Slika 9

Upravo ste naučili kreirati projekt i napraviti pull request na GitHubu.

3. Zadatak za vježbu

- Kreirati novi javni repozitorij i u njega dodati osobe iz tima.
- Napraviti jednostavan HTML dokument te ga uploadati na GitHub.
- Kreirati novu granu te napraviti izmjene u dokumentu.
- Commitati i uočiti promjene.
- Dodijeliti zadatke kolegama iz tima po dogovoru.