

GIT

Šta je Git?

Git je besplatan open source sistem kontrole verzija osmišljen da rukovanje malim ili velikih projekatima sa fokusom na brzinu i efikasnost.

Git prati sve promjene koje se dešavaju na projektu na računaru svakog pojedinačnog člana tima i sinhronizuje ih na udaljeni zajednički repozitorijum. Na ovaj način programerima je omogućeno da uvijek imaju pristup poslednjoj verziji projekta. Zbog toga se za git kaze da služi i za verzionisanje koda.

Šta je Git?

- Predstavlja platformu za hostovanje i kolaboraciju na projektu,
- Sistem za kontrolu verzija fajlova ,
- Sinhronizacija i upravljanje ,
- Omogućava izradu boljeg softvera kroz visok stepen kolaboracije

Osnovni pojmovi

- Repositories,
- Branches,
- Commits,
- Issues i
- Pull Requests

Prednosti upotrebe Git-a

- Svako dobija svoju kopiju projekta što znači da je Git vrlo brz i radi lokalno (offline).
- Promene koda i spajanje promena .
- Developeri modifikuju kod .
- Developeri dodaju , pomeraju, brišu fajlove

Instalacija Git klijenta

Softver koji se zove GitBash. Ovo je softver za komunikaciju sa GitHubom korišćenjem konzolnih komandi. Vrlo je važno prvo naučiti konzolne komande, a tek potom koristiti aplikacije sa grafičkim korisničkim interfejsom.

Možete preuzeti sa adrese <https://git-scm.com/downloads>

The screenshot shows the official Git website at <https://git-scm.com/>. The main navigation bar includes links for About, Documentation, Downloads, and Community. The Downloads section is highlighted, showing links for Mac OS X, Windows, and Linux/Unix. A large image of a Mac monitor displays the latest source release version 2.19.2. Below the download links, there's information about older releases and a link to view GUI clients. The page also features sections for GUI Clients and Logos, along with social media sharing icons.

git --distributed-is-the-new-centralized

Search entire site...

About

Documentation

Downloads

GUI Clients

Logos

Community

Downloads

Mac OS X Windows

Linux/Unix

Latest source Release
2.19.2
Release Notes (2018-11-21)

Download 2.19.0 for Mac

Older releases are available and the Git source repository is on GitHub.

GUI Clients

Logos

View GUI Clients →

Git via Git

If you already have Git installed, you can get the latest development version via Git itself:

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

You can also always browse the current contents of the git repository using the web interface.

Instalacija Git klijenta

[Download git za OSX](#)

[Download git za Windows](#)

[Download git za Linux](#)

The screenshot shows the official Git website at git-scm.com. The top navigation bar includes links for "About", "Documentation", "Downloads", and "Community". The "Downloads" section is highlighted. It features links for "Mac OS X", "Windows", and "Linux/Unix". A large image of a Mac desktop monitor displaying the Git logo is on the right. Below the download links, there's information about the "Pro Git book" and links to "View GUI Clients" and "Logos".

Latest source Release
2.19.2
Release Notes (2018-11-21)

Download 2.19.0 for Mac

Mac OS X Windows Linux/Unix

Older releases are available and the [Git source repository](#) is on GitHub.

GUI Clients
Git comes with built-in GUI tools (`git-gui`, `gitk`), but there are several third-party tools for users looking for a platform-specific experience.
[View GUI Clients →](#)

Logos
Various Git logos in PNG (bitmap) and EPS (vector) formats are available for use in online and print projects.
[View Logos →](#)

Git via Git
If you already have Git installed, you can get the latest development version via Git itself:
`git clone https://github.com/git/git`
You can also always browse the current contents of the git repository using the web interface.

Git Hub

GitHub

- online cuvanje koda, 'drustvena mreža' za programere



Built for developers

GitHub is a development platform inspired by the way you work. From [open source](#) to [business](#), you can host and review code, manage projects, and build software alongside 31 million developers.

Username

Pick a username

Email

pajevicn@gmail.com

Password

Make sure it's more than 15 characters, or at least 8 characters, including a number, and a lowercase letter.

Sign up for GitHub

By clicking "Sign up for GitHub", you agree to our [terms of service](#) and [privacy statement](#). We'll occasionally send you account related emails.

Prvo podešavanje

Prva stvar koju treba da uradite kada instalirate Git je da podesite korisničko ime i mejl adresu. Ovo je važno jer svaki Git komit koristi ove podatke, i nepromjenljivo se urezuje u komite koje kreirate:

```
$ git config --global user.name "Nikola Pajevic"  
$ git config --global user.email nikolapajevic@primjer.com
```

Provjera

Ako želite da provjerite podešavanja, možete da koristite git config --list komandu da izlistate sva podešavanja koja Git može da nađe u tom trenutku:

```
$ git config --list
user.name=Nikola Pajevic
user.email=nikolapajevic@example.com
color.status=auto
color.branch=auto
color.interactive=auto
color.diff=auto
...
...
```

Traženje pomoći

Ako vam ikad bude bila potrebna pomoć dok koristite Git, postoje tri načina da dođete do stranice sa upustvom (manual page i skraćeno manpage) za bilo koju od Gitovih komandi:

```
$ git help <glagol>  
$ git <glagol> --help  
$ man git-<glagol>
```

Osnovna upotreba

- git init
- git status
- git add .
- git commit -m "opis promjena"
- git log

Dodavanje lokalne grane

Neophodno je da prije izvršavanja naredbe, u komandnoj liniji namjestimo putanju koja ukazuje na folder koji želimo da pratimo.

Komandom:

cd (link foldera)

Mi imamo lokalni(local) i udaljeni(remote) repozitorijum. Pravljenje repozitorijuma se vrši sa naredbom u komandnoj liniji:

git init

Pomoću te naredbe, mi pravimo master granu na svom lokalnom repozitorijumu. To je glavna grana na kojoj se nalazi sav naš kod. Mogu se praviti nove grane za nove potrebe.

```
USER@USER-PC MINGW64 ~
$ cd /c/Users/USER/Desktop/PROJEKTI/website

USER@USER-PC MINGW64 ~/Desktop/PROJEKTI/website
$ git init

USER@USER-PC MINGW64 ~/Desktop/PROJEKTI/website (master)
$
```

Git status

Komandom git status provjeravamo stanje naših fajlova i promjena na njima. Ako se promijenilo stanje na našoj radnoj verziji, Git će istaći fajlove koji su izmijenjeni. Biće istaknuti crvenom bojom.

```
USER@USER-PC MINGW64 ~/Desktop/PROJEKTI/website (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

    deleted:   .gitignore

no changes added to commit (use "git add" and/or "git commit -a")

USER@USER-PC MINGW64 ~/Desktop/PROJEKTI/website (master)
$
```

Git add

Nakon dodavanje fajlova možemo ponovo koristiti komandu git status da bismo provjerili stanje na nasoj radnoj verziji. Novi dodati fajlove biće istaknuti zelenom bojom.

Slanje određenog fajla

Slanje samo određenog fajla u pripremni prostor sa naredbom:
git add (naziv fajla koji dodajete)

```
USER@USER-PC MINGW64 ~/Desktop/PROJEKTI/website (master)
$ git add webweb.txt
```

Slanje svih fajlova:

Ako želimo da dodamo sve fajlove koristimo komandu:

git add -A

```
USER@USER-PC MINGW64 ~/Desktop/PROJEKTI/website (master)
$ git add -A

USER@USER-PC MINGW64 ~/Desktop/PROJEKTI/website (master)
$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

            renamed:   .gitignore -> webweb.txt
```

Git commit

Pomoću git commit-a pravimo snimak promjena stanja fajlova, koje smo prethodno izabrali. Zapravo, sačuvamo sve što smo do tada odradili i šaljemo poruku u kojoj navodimo šta smo uradili. Snimke promjena uvek commitujemo samo u svoj lokalni rezervorijum na računaru. To odrađujemo komandom:

```
git commit -m "poruka kojom objašnjavamo šta smo odradili"  
-m(message)
```

Povezivanje

Da bismo zadali komandu pomoću koje će se povezati lokalni i udaljeni repozitorijum, moramo imati udaljeni repozitorijum na koji će da se poveže lokalni. Zato na GitHub-u pravimo novi repozitorijum.

The screenshot shows two side-by-side sections of the GitHub interface. On the left, the 'Create a new repository' form is displayed. It includes fields for 'Owner' (set to 'kajaned24'), 'Repository name' (set to 'Website'), a 'Description (optional)' field, and options for 'Public' or 'Private' visibility. A note says 'Great repository names are short and memorable. Need inspiration? How about symmetrical-octo-train.' Below these are checkboxes for initializing with a README and adding .gitignore and license files. At the bottom is a green 'Create repository' button. On the right, a 'Quick setup' guide is shown. It starts with a 'Set up in Desktop' section showing connection options (HTTPS, SSH) and a link to 'https://github.com/kajaned24/Website.git'. An annotation with an arrow points to this link. Below it, instructions for creating a new repository on the command line are provided, along with a code block:

```
echo "# Website" >> README.md  
git init  
git add README.md  
git commit -m "first commit"  
git remote add origin https://github.com/kajaned24/Website.git  
git push -u origin master
```

Further down, instructions for pushing an existing repository from the command line are given, with another code block:

```
git remote add origin https://github.com/kajaned24/Website.git  
git push -u origin master
```

At the bottom, there's an 'Import code' button and a note about importing code from other repositories.

Kreirajte novi folder, otvorite ga i izvršite sledeću komandu: git init tako ste kreirali novi git repozitorijum

Git remote add origin

Remote je udaljeni repozitorijum koji ima ime(koje se kasnije koristi da bi koristili komandu push i pull) i url(koji je zadužen da odredi na kojoj se adresi nalazi taj repozitorijum). Da ne bismo svaki put pisali url dovoljno je da kažemo „origin“ i lokalni repozitorijum zna gdje da šalje.

Remote može da ima bilo koje ime ali je konvencija(dogovor među globalnom IT zajednicom) odredila da se remote zove origin.

Komanda:

git remote add origin (link udaljenjog repozitorijuma)

Git push

Git se koristi za rad u timu da bi programeri mogli na laksi način da rade na istom projektu. Zbog toga, ako radimo u timu, moramo prvo odraditi komandu pull(povući), ukoliko se na udaljenom repozitorijumu nalazi nešto što nemamo u našem lokalnom, da bismo mogli da odradimo push(„gurnuti”).

Ako nemamo ništa na udaljenom repozitorijumu odmah radimo komandu push. Ukoliko smo dobro povezali lokalni sa udaljenim repozitorijumom a imamo uradjen bar prvi commit, možemo da pošaljemo sve fajlove sa lokalnog na prazan udaljeni repozitorijum sa naredbom:

```
git push origin master
```