

Uvod u Git i GitHub

Marko Jeremic

Zasto git a ne dropbox?

- Git cuva sve verzije fajla, a ne samo naj noviju
- Git ne sync-uje automatski, i sync-uje samo stvari koje odaberete
- Git omogućava da imate vise verzija vas eg koda na jednoj lokaciji

Kako git repository izgleda?



Kako izgleda commit?

main.c

```
1 #include <stdio.h>
2
3 int main()
4 {
5     printf("Hello World");
6     return 0;
7 }
```



main.c

```
1 #include <stdio.h>
2
3 int main()
4 {
5     printf("Good Morning");
6     return 0;
7 }
```

commit:

```
5 --- printf("Hello World");
5 +++ printf("Good Morning");
```

Kako izgleda commit za binarne fajlove?

video.mp4

```
1 101101011010101010100101101  
2 101101011010101010100101101  
3 101101011010101010100101101  
4 101101011010101010100101101  
5 101101011010101010100101101  
6 101101011010101010100101101  
7 101101011010101010100101101
```



video.mp4

```
1 101101011010101010100101101  
2 101101011010101010100101101  
3 101101011010101010100101101  
4 101101011010101010100101101  
5 101101011010101010100101101  
6 101101011010101010100101101  
7 101101011010101010100101101
```

commit:

????????????


Sajtovi za hostovanje Git repositorijuma

- Najpoznatiji sajtovi za git su GitHub i Bit Bucket
- Jedina razlika izmedju funkcionalnosti git hosting sajtova jeste njihov web interfejs, ono sto mozete da odradite u git terminalu sa jednim, mozete i sa drugim




Konfigurisanje Git-a

- Dve neophodne konfiguracije su:
 - `git config --global user.name "username"`
 - `git config --global user.email "email@gmail.com"`
- Bonus konfiguracije:
 - `git config --global core.editor nano`
 - `git config --global core.autocrlf input`

Kreiranje repo-a preko GitHub-a



[Pull requests](#) [Issues](#) [Gist](#)


  

Create a new repository

A repository contains all the files for your project, including the revision history.


Owner


Repository name

 cra0zy ▾ /

Great repository names are short and memorable. Need inspiration? How about **curly-rotary-phone**.


Description (optional)

☒  **Public**
Anyone can see this repository. You choose who can commit.

☐  **Private**
You choose who can see and commit to this repository.

☐ **Initialize this repository with a README**
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** ▾

Add a license: **None** ▾ 

Create repository

Downloadovanje (kloniranje) Git Repoa

- Da klonirate repo odradite (ovo downloaduje s ve grane i sve commit-ove):
 - `git clone <URL> <*Name>`
- Ako hocemo da kloniramo samo povrсни sloj mozemo da dodamo depth flag:
 - `git clone <URL> <*Name> --depth 1`

Dodavanje i Commit-ovanje Fajlova

- *git status* - pokazuje trenutni status fajlova
- *git add <file>* - dodaje fajl u list fajlova spremnih za commit-ovanje
- *git commit -m "message"* - kreiranje commita koristeći dodate fajlove

Brisanje i Uncommit-ovanje Fajlova

- *git checkout <file>* - ova komanda radi samo ako fajl nije na dodatoj listi fajlova, resetova ce fajl nazan na njegovo prethodno stanje
- *git reset <file>* - uklanja fajl iz list fajlova spremnih za commit-ovanje
- *git reset HEAD~* - uklanja poslednji commit

Koriscenje .gitignore Fajla

- .gitignore je samo uobicajan text fajl sa listom fajlova koje git treba da zaobidje
- Svi fajlovi koje git zaobilazi mogu biti obrisani sa jednostavnom komandom:
- *git clean -fdx*

Primer .gitignore Fajla

Terminal:

```
[user]$ ls -a  
.  ..  .git bin main.c main.c~ .gitignore
```

.gitignore:

```
bin/
```

```
*~
```

Syncovanje Repo-a

- Kad napravite promene one nece automatski da se uploaduju/downloaduju
- Da uploadujete sve promene odradite:
 - *git push*
- Da downloadujete sve promene odradite:
 - *git pull*

Grane (branches)

- Najcesse koriscene grane su master i develop
- git branch - lista trenutne grane i pokazuje granu na kojoj ste
- git branch <name> - kreira novu granu sa zadatim imenom
- git checkout <name> - prelazi na granu sa zadatim imenom

Spajanje Grana

- *git merge <branch>* - dodaje sve commit-ove iz grane na sam kraj
- *git rebase <branch>* - prvo postavlja commit-ove iz druge grane, a zatim redom p ostavlja commit-ove iz sebe
- kada dva commit-a modifikuju istu linij u i vi pokušate da ih spojite u jednoj grani dogodice se *merge conflict*

Forkovanje (kopiranje)

- Kopiranje nekog reposetorijuma se naziva forkovanje
- Ovo radite ako želite da napravite promene reposetorijumu koji nije vas
- Submitovanje promena koje želite da napravite originalnom repo-u se naziva *pull request*

Remote Hosts

- Najcesce korisceni hostovi su *origin* i *upstream*
- *git remote -v* - lista sve poznate hostove
- *git remote add <name> <url>* - dodaje novi remote host
- *git remote remove <name> <url>* - brise remote host
- *git fetch <name>* - downloaduje commit-ove od hosta

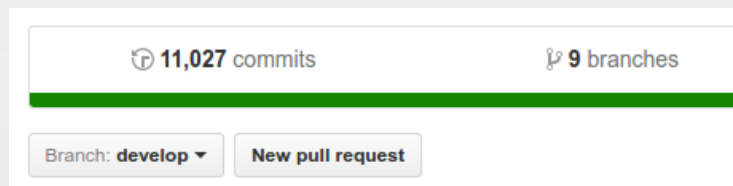
Syncovanje Forkova

Terminal:

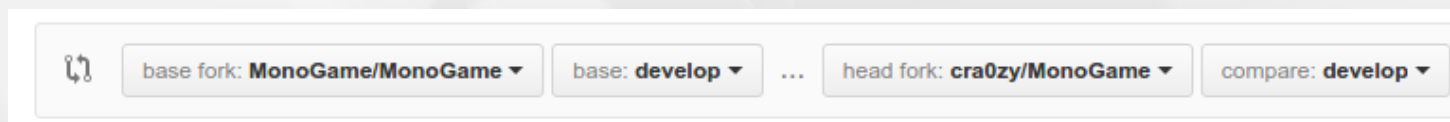
```
[user]$ git remote add upstream <url>
[user]$ git fetch upstream
[user]$ git rebase upstream/<branch>
[user]$ git push <branch> --force
[user]$ _
```

Submitovanje Pull Requestova

- Kliknite na "New pull request" dugme u gornjem levom uglu na web strani vaseg reposetorijuma



- Selektuje gde zelite (levo) i koju granu (desno) da trazite da budu spojene i kliknite "Create Pull Request"



Rebasing

- Rebasovanje vam omogucava da modifikujete committe direktno (spojite ih, brisete ih, editujete te ih, etc.)

git rebase -i
HEAD~N

The End

```
[user]$ git add .
```

```
[user]$ git commit -m "The End"
```

```
[user]$ git push
```

```
[user]$ _
```