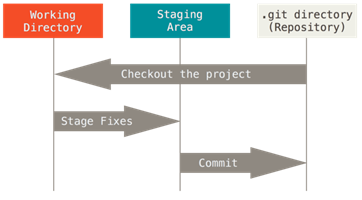
Git & Github

To je alat za pravljenje “verzija“ aplikacije. U praksi to znaci da za svaku manju ili vecu izmjenu u fajlovima mozemo imati stanje kreirano kroz komite (commits) na koje se mozemo vratiti.

Github je distribuirani sistem (server u sustini) koji prati i sinhronizuje lokalne git verzije aplikacije. U odredjenom smislu, github je bakcup verzija aplikacije, kao sto su takodje lokane vezije backup server vezije. U slucaju da jedna strana „padne“, druga ostaje. Jos veci benefit se ogleda u mogucnosti nezavisnog rada developera, odnosno tima, kroz zasebne grane, pri cemu svaki developer radi na odredjenoj funkcionalnosti ne ometajuci druge clanove tima (cak I ako se ovo desi, postoji mogucnost odstranjivanja konflikata pri cemu developer analizira svoje I tudje izmjene I bira koji kod ce zadrzati).

Za upotrebu gita, neophodno je instalirati aplikaciju sa sledece adrese:<https://git-scm.com/download/win>.

Za svaki projekat za koji koristimo git, kreira se skriveni .git folder. U ovom folderu se zapravo „prate“ sve lokalne izmjene. Izmedju projekata i ovo foldera postoji „prelazna staging“ faza, u koju se smjestaju fajlovi prije slanja u git. To izgleda ovako.



Najbitnije git komande:

- git help <komanda> – pomoc u vezi neke komande

- git init – inicijalizuje lokalni git projekat

- git add . – dodaje sve fajlove na kojima je bilo izmjena ili su kreirani u „staging“ prostor

- git commit -m „neka poruka“ – potvrde lokalnih izmjena i slanje istih u lokalni git

- git clone putanja-do-eksternog-repositorija – kloniranje postojeceg projekta sa gita

- git status – ispisuje lokalne izmjene fajlova

- git diff – prikazuje fajlove koji su mijenjani, ali nisu jos u „stage“

- git rm naziv-fajla – za uklanjanje fajla iz gita

- git commit --amend – dodatno dopunjavanje komita koji je vec u „stage“

- git restore naziv-fajla – odbacuje izmjene iz nekog fajla

- git fetch – „povlaci“ izmjene is externog repozitorija, ali ih ne sjedinjava direktno sa lokalnom verzijom

-git pull - „povlaci“ izmjene is externog repozitorija, i sjedinjava ih direktno sa lokalnom verzijom

- git push origin naziv-udaljene-grane – sinhronizuje lokalne komitovane izmjene sa remote serverom

- git checkout naziv-grane – kreira novi branch i prebacuje nas na njega

- git switch naziv-grane - prebacuje na drugi branch, ali ne kreira ako ne postoji

- git branch naziv-grane – kreira novu granu

Sta je branch u stvari? To je u sustini pointer, koji je izvezen iz trenutnog stanja osnovne grane iz koje je kreiran i koji ima svoje odvojeno stanje u odnosu na ostale grane. Ono sto je bitno, jeste da se izmjene na odredjenoj grani ne vide na drugim granama do trenutka dok se ne odradi sparanje (merge) grana.

.gitignore fajl sluzi za ignorisanje izmjena na nekim od lokalnih fajlova (npr: .env fajl)