Git submoduli

mogućnost korištenja repozitorija unutar repozitorija

Dario Barać, Anđelo Ferenčić

Uvod u git submodule

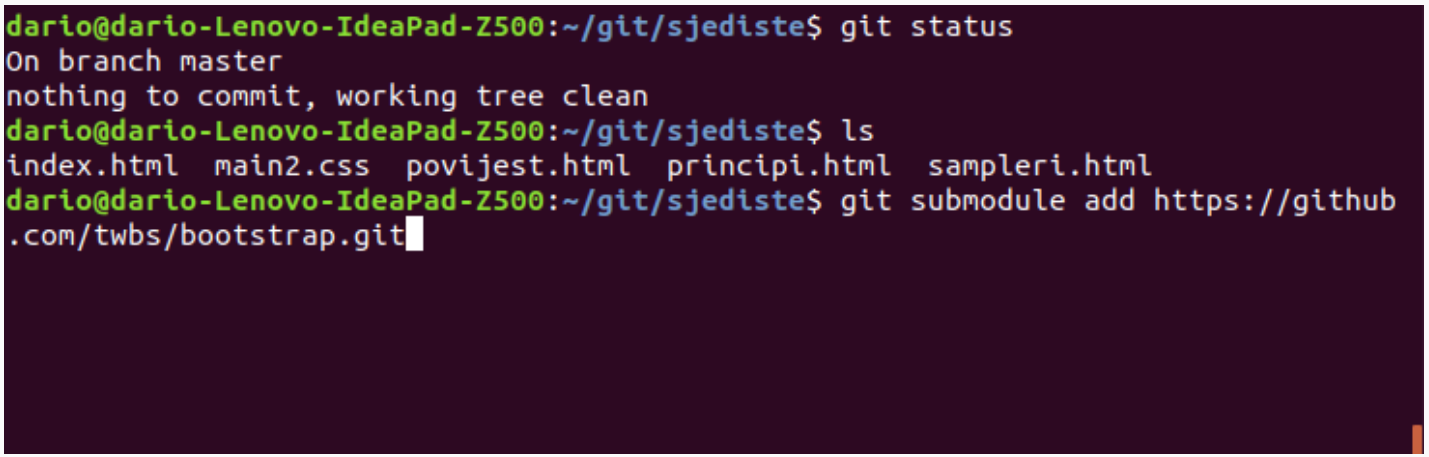
- Ako razvijamo projekt unutar kojeg se nalazi drugi projekt (npr. Bootstrap library unutar repozitorija web sjedišta) može doći do problema sa praćenjem njihovih verzija.
- Git taj problem riješava sa submodulima koji nam omogućuju njihovo neovisno verzioniranje.

Početak rada sa submodulima

U repozitoriju našeg web sjedišta ćemo inicijalizirati git submodul za Bootstrap library koristeći naredbu:

```
$submodule add
```

Submodul će biti spremljen u poddirektorij projekta i imat će isti naziv kao originalni repozitorij kojeg uključujemo.

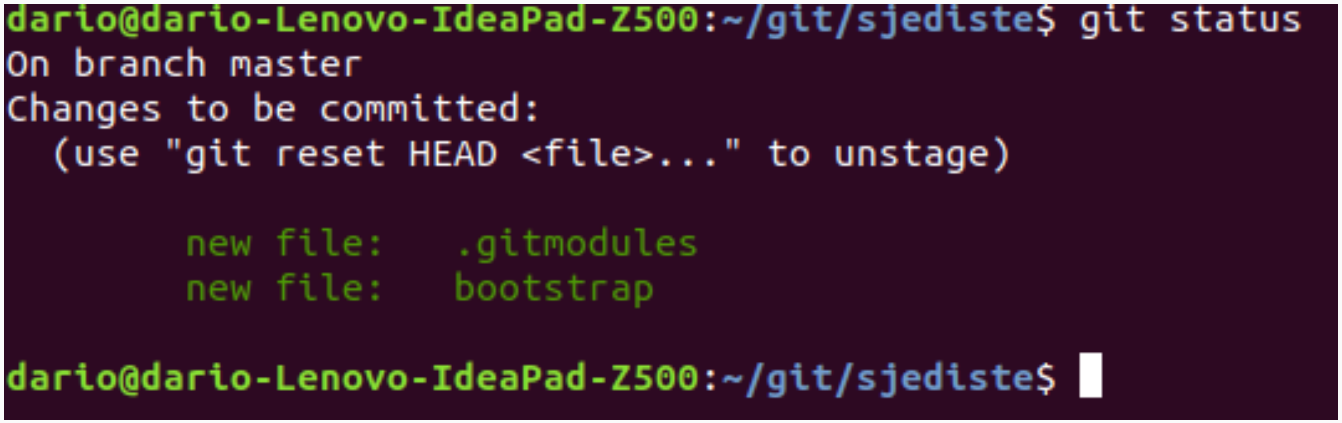


Početak rada sa submodulima

Naredba

```
$git status
```

nam javlja daje stvorena .submodule datoteka koja sadrži imena i url adrese svih submodula.



Možemo primjetiti i da git ne prati *bootstrap*-ove datoteke jer zna da je to podrepozitorij.

Početak rada sa submodulima

I kada napravimo *commit* git prati podrepozitorije kao samo jednu datoteku.

```
dario@dario-Lenovo-IdeaPad-Z500:~/git/sjediste$ git commit -m "Dodan Bootstrap s
ubmodul"
[master 6650dbe] Dodan Bootstrap submodul
 2 files changed, 4 insertions(+)
 create mode 100644 .gitmodules
 create mode 160000 bootstrap
dario@dario-Lenovo-IdeaPad-Z500:~/git/sjediste$
```

Kloniranje submodulima

Kada kloniramo repozitorij dobijemo dobijemo i prazan direktorij njegovih submodula.
Sljedeće naredbe dohvaćaju sve datoteke koje su potrebne za rad podrepozitorija:

```
$git submodule init
```

```
$git submodule update
```

Kloniranje submodulima

Brža alternativa za to je da kod kloniranja dodamo i:

```
--recurse-submodules
```

```
dario@dario-Lenovo-IdeaPad-Z500:~/git$ git clone sjediste sjediste3 --recurse-su
bmodules
Cloning into 'sjediste3'...
done.
Submodule 'bootstrap' (https://github.com/twbs/bootstrap.git) registered for pat
h 'bootstrap'
Cloning into '/home/dario/git/sjediste3/bootstrap'...
remote: Counting objects: 119170, done.
remote: Compressing objects: 100% (66/66), done.
remote: Total 119170 (delta 95), reused 103 (delta 80), pack-reused 119022

Receiving objects: 100% (119170/119170), 112.16 MiB | 2.06 MiB/s, done.
Resolving deltas: 100% (79683/79683), done.
Submodule path 'bootstrap': checked out '96a9e98e49cdd32148981ef71617b1cfc605447
0'
dario@dario-Lenovo-IdeaPad-Z500:~/git$
```

Rad na projektima sa submodulima

Sadržaj podrepozitorija u našem projektu možemo ažurirati tako da unutar njihovih direktorija napišemo ove naredbe:

```
$git fetch
```

```
$git merge
```

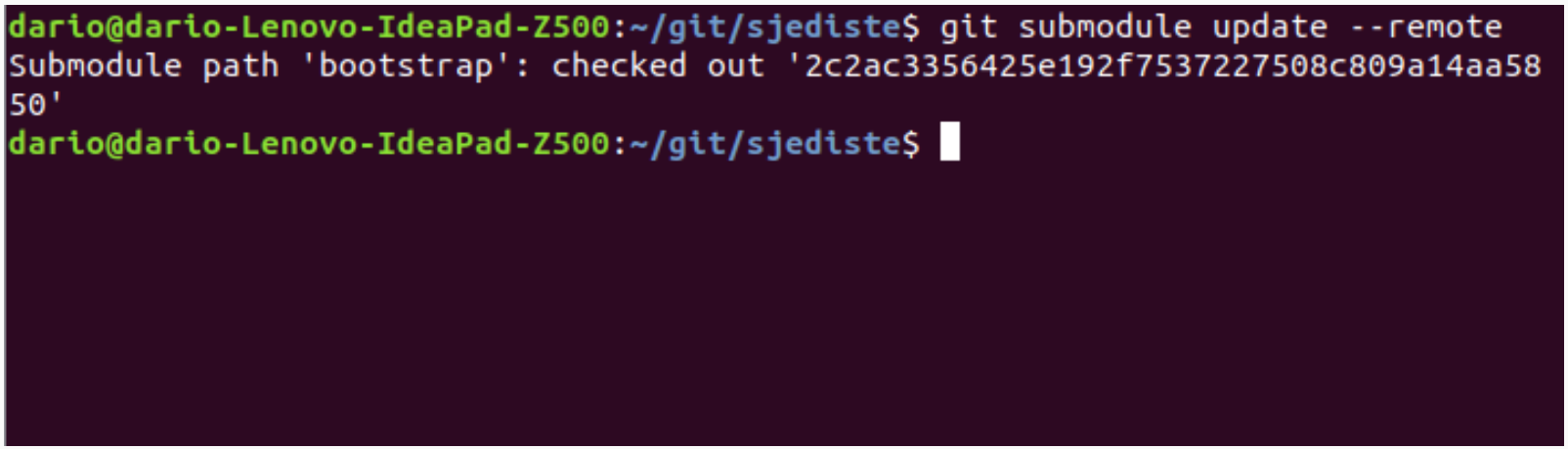
Ako imamo više podrepozitorija u projektu, ažuriranje nam može oduzeti dosta vremena.

Rad na projektima sa submodulima

To se rješava u glavnom direktoriju projekta sljedećom naredbom:

```
$git submodule update --remote
```

Time smo ažurirali sve submodule u našem projektu.



Rad na projektima sa submodulima

`$git status`

U glavnom direktoriju javlja da imamo nove *commitove* u podrepozitorijima.

```
dario@dario-Lenovo-IdeaPad-Z500:~/git/sjediste$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   bootstrap (new commits)
```

`$git diff --submodule`

Pokazuje koji su to *commitovi*:

```
dario@dario-Lenovo-IdeaPad-Z500:~/git/sjediste$ git diff --submodule
Submodule bootstrap 96a9e98e4...2c2ac3356:
  < Update devDependencies. (#25371)
  < Feature: Add carousel fade option (#22958)
  < v4.1: Add .dropdown-item-text (#22965)
```

Rad na submodulima

Koristeći naredbu

```
$git submodule update
```

Git pribavlja promjene i ažurira datoteke u poddirektoriju, no nijedna grana ne prati naše promjene u podrepozitoriju.

Rad na submodulima

Da bismo riješili taj problem moramo napraviti dvije stvari:

- Ući u svaki submodul i "*checkout*"-at granu na kojoj ćemo raditi.
- Konfigurirati što će Git napraviti ako smo napravili promjenu te onda naredbom

  ```
  $git submodule update --remote
  ```

  pullamo nove promjene.

Rad na submodulima

Dodavanjem

`--merge`

na korišteni git update poziv spojit ćemo promjenu koja je na serveru za ovaj submodul sa našom granom.

```
$git submodule update --remote --merge
```

Rad na submodulima

Dodavanjem

`--rebase`

spojit ćemo promjenu našu lokalnu promjenu sa serverom.

Objavljivanje promjena na submodulima

Ako imamo promjene na glavnom projektu i lokalne promjene u submodulu i commitamo glavni projekt i pushamo ga bez pushanja promjena na submodulu ostali koji rade na projektu neće imat pristup promjenama na submodulu.

Objavljivanje promjena na submodulima

Da bismo ovaj problem riješili na naredbu

```
git push
```

nadodamo argument

```
--recurse-submodules
```

koji može biti "check" ili "on-demand".

Objavljivanje promjena na submodulima

"check" opcija samo javi da promjene na submodulu nisu pushane te zaustavi push. "on-demand" opcija pokuša pushati promjene na submodulima i nakon toga glavni projekt.

Spajanje promjena submodula

U slučaju spajanja promjena u isto vrijeme kad i netko drugi, može doći do konflikata.

Foreach naredba

Ako imamo više submodula u istom projektu naredba foreach olakšava stvari. Npr. ako izradimo novu granu i želimo se prebaciti na nju na svim submodulima.

Problemi sa submodulima

Grananje sa submodulima može biti problematično. Ako napravimo novu granu, dodamo submodul na tu granu i vratimo se na granu bez tog submodula još uvijek imamo direktorij submodula kao nepraćeni direktorij.

Problemi sa submodulima

Ako izbrišemo taj direktorij, nakon što se ponovo vratimo na granu sa submodulom moramo pozvati naredbu

```
git submodule update –init
```

da bi ponovo je popunili.

Problemi sa submodulima

Mijenjanje iz poddirektorija u submodule je također jedan od čestih problema. Mora se unstageat poddiretorij i tek onda se može dodati submodul.

Literatura


S. Chacon and B. Straub, *Pro Git*.
The expert's voice, Apress, 2014.