

Operare con Git su Netbeans:

Creare un repository:

- tasto destro sul progetto\versioning\Initialize Git Repository

Effettuare un commit

- tasto destro sul progetto\Git\Commit

Effettuare un checkout

- tasto destro sul progetto\Git\Checkout|Checkout revision>Select
- Ora si possono scegliere i branch e i commit da cui effettuare il checkout
ATTENZIONE: Se fai un checkout è sempre meglio farlo creando un nuovo branch altrimenti non è più possibile effettuare checkout su commit “avanti” una volta che sei andato indietro!

Creare un nuovo branch

- Git\(\Branch\Tag)\Create Branch

Spostarsi su un altro Branch/Revision

- Git\(\Branch\Tag)\Switch to Branch

Fare un merge

- Portarsi con Head sulla revision di destinazione ad esempio “master\ultimo commit”
- Git\(\Branch\Tag)\Merge revision
- Selezionare la revision da unire al Head, ad esempio “sviluppo\ultimo commit”
- Selezionare la spunta su fast Forware if possible)
- Selezionare Merge

In caso di eventuali conflitti viene mostrata una finestra: scegliere “Resolve per risolvere manualmente i conflitti”

Si apre una finestra “Merge Conflict Resolver” in cui vengono mostrate le due versioni da unire, con le parti differenti evidenziate, come mostrato di seguito:

Revision puntata dal HEAD

Revision da "mergiare" con la revision HEAD

Risultato del Merge modificabile cliccando sui pulsanti "Accept, Accept Both". Non è possibile modificare "a mano".

Conflict: 1 of 1, Unresolved: 0

Accept Accept Both Accept & Next

```
37 // Scanner tastiera=new Scanner(System.in);
38 ConsoleInput tastiera=new ConsoleInput();
39 String nomeFileCSV="libri.csv";
40 String nomeFileBin="libri.bin";
41
42* System.out.println("LIBRERIA BRANCH refactoring successivi");
43 String vociMenu[]=new String[11];
44 {
45     vociMenu[0]="Esci";
46     vociMenu[1]="Visualizza libri presenti nello scaffale";
47     vociMenu[2]="Aggiungi volume";
48     vociMenu[3]="Elimina volume (ripiano, posizione)";
49     vociMenu[4]="Cerca volume (ripiano, posizione)";
50     vociMenu[5]="Cerca libri autore";

```

HEAD (41ff7c2) - Aggiunto titolo del sw

Accept Accept Both Accept & Next

```
37 // Scanner tastiera=new Scanner(System.in);
38 ConsoleInput tastiera=new ConsoleInput();
39 String nomeFileCSV="libri.csv";
40 String nomeFileBin="libri.bin";
41
42* System.out.println("PROGETTO LIBRERIA NUOVO BRANCH");
43 String vociMenu[]=new String[11];
44 {
45     vociMenu[0]="Esci";
46     vociMenu[1]="Visualizza libri presenti nello scaffale";
47     vociMenu[2]="Aggiungi volume";
48     vociMenu[3]="Elimina volume (ripiano, posizione)";
49     vociMenu[4]="Cerca volume (ripiano, posizione)";

```

a6500d328bb34b74fb10674ab23ed843dd33ff94

Result after merge

App.java

Nome del file oggetto di conflitto

Clonare Repository da Github con NetBeans

Un repository GIT locale può essere caricato sulla piattaforma GitHub e quindi clonato da chiunque.

Dall'IDE NetBeans (selezionando un qualsiasi progetto per il quale non sia ancora stato inizializzato un repository GIT oppure chiudendo tutti i progetti)

- Team/Git/Clone
- Inserisci URL repository remoto su gitHub
- Specifica il path in cui clonare la repository in locale
- Next
- Seleziona quali branch clonare dal repository remoto
- Next
- Seleziona il Branch sul quale si vuole effettuare il checkout
- Finish

Per fare il checkout su altri branch del repository scaricato

Tasto destro sul progetto:

- Git\checkout
- fare il checkout sul branch desiderato del repository remoto (selezionando checkout as new branch)

Creare il progetto NetBeans e caricarlo su gitHub

1. Creare il proprio progetto su netBeans
2. creo repository locale con: Team\Git\Initialize repository...
3. Fai un commit con tasto dx sul progetto:Git\commit
(verrà creato un branch master)
4. Vai su GitHub
5. Crea un repository con lo stesso nome del repository locale
6. Aggiungi un file README in cui descrivi il progetto con una frase
7. Prima di fare il commit, se necessario, puoi cambiare il nome del branch, rinomina il branch "master" se non si chiama già così.
8. Fai il commit su GitHub.
9. Copia l'URL del repository remoto
10. Torna sull'IDE NetBeans
11. Fai un pull con tasto dx sul progetto Git\Remote \pull
12. Seleziona "Specify Git Repository Location", incolla l'URL del repository remoto (il suo branch si chiamerà Origin)
13. Non è necessario inserire User e Password
14. Next
15. Seleziona il branch remoto (origin/master) che si vuole "mergiare" con il branch locale master
16. Ti chiederà se fare un merge, fallo.
17. Ora il repository locale conterrà anche i file del repository su GitHub (il file README.txt), verifica che ci sia. Si sottolinea che il termine **origin indica il repository su GutHub.**
18. Ora bisogna fare un push su GitHub del progetto: tasto destro sul progetto\Git\Remote\Push
19. Verifica che l'url indicato in "Select Configured Git Repository Location:" sia quello di origin su GitHub poi premi Next
20. Seleziona quali branch del repository locale inviare su origin
21. Clicca Next e Finish
22. Inserisci la tua userName e PW per avere il permesso di eseguire il push,
ATTENZIONE: la PW non è la stessa usata per l'accesso a GitHub, è un TOKEN (una lunga stringa alfanumerica) che bisogna generare su GitHub per avere la possibilità di usarne le WebAPI (evitare di memorizzare la password). Per ottenere il TOKEN bisogna:
 - a. accedere al proprio account GitHub
 - b. cercare Personal Access Token
 - c. Generare il TOKEN
 - d. Salvarlo su un proprio file personale per riutilizzarlo ogni volta che si vuole effettura un push da NetBeans

Personal access tokens (classic) Generate new token Revoke all

Tokens you have generated that can be used to access the [GitHub API](#).

Settembre 2024 — repo	Last used within the last week	Delete
-----------------------	--------------------------------	--------

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

23. Fai un push.

24. Fatto. Verifica che su GitHub sia presente il branch con lo stesso commit del repository locale.

Le istruzioni dalla numero 18 in poi consentono di fare successivi Push su GitHub.

COSA SONO SU FORK E PULL REQUEST SU GITHUB

Fork: un fork è un **clone** di una repository di qualcun altro sul mio account github

Pull request è una richiesta, da parte di un altro utente, di effettuare un merge del suo fork alla mia repository diventando quindi un collaboratore con il permesso di effettuare commit direttamente sulla mia repository, vedi qui:

https://www.youtube.com/watch?v=tHzHXtSOxqo&ab_channel=OODeveloper