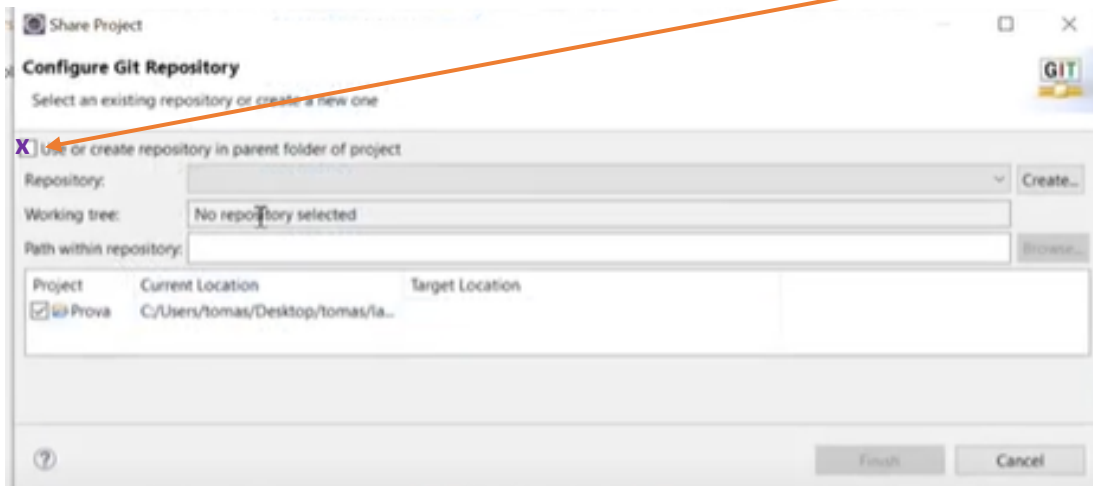
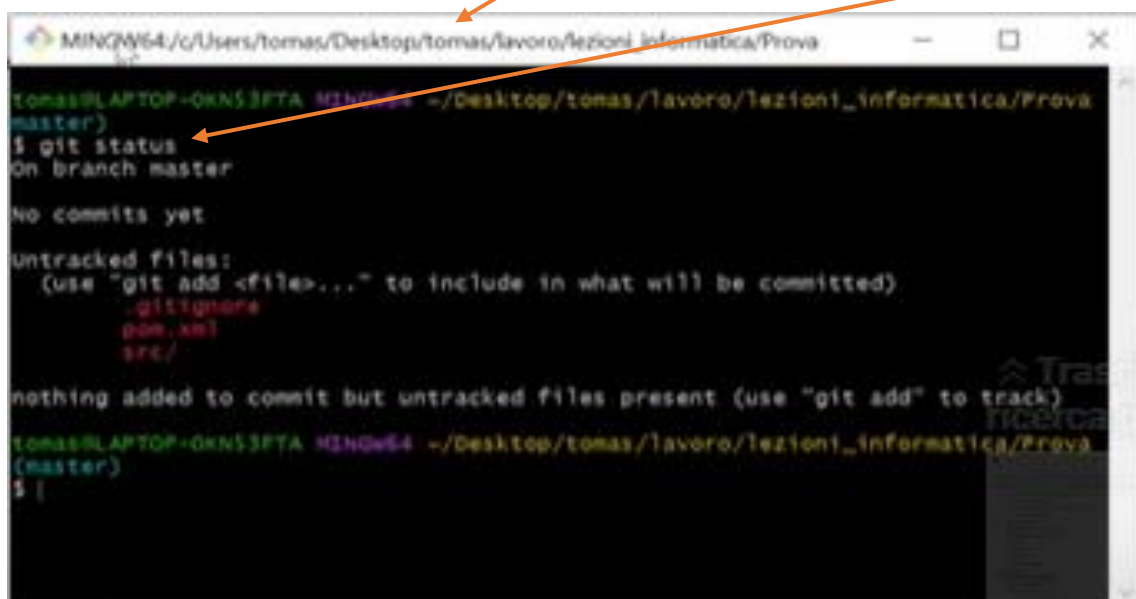


Come aggiungere un repository da Eclipse su git hub:

1. Per prima cosa aprire git e creare un nuovo repository.
2. Come prima cosa andare sul progetto tasto destro → **Team**
3. Selezionare → **Share Project**
4. Si aprirà questa finestra di dialogo, dobbiamo spuntare questa → **X**



5. Successivamente cliccare dove appare → **create repository**, poi → **Finish**, questa operazione inizializza git, infatti se andremo nella cartella dove abbiamo il nostro progetto troveremo una cartella git, nella cartella principale del progetto manca il file gitignore, copiarlo da un altro progetto e incollarlo qui, serve → (Quando facciamo la build del progetto di default finirà nella cartella target, così questa parte non verrà pushata su git).
6. Successivamente nella cartella del progetto (nello spazio bianco) tasto destro e selezionare → **Git Bush Her**, successivamente si aprirà la finestra di testo, dove digiteremo → **git status**, così da poter visualizzare lo stato.



```
tomas@LAPTOP-OKN53FTA MINGW64 ~/Desktop/tomas/lavoro/lezioni_informatica/Prova
(master)
$ git status
On branch master

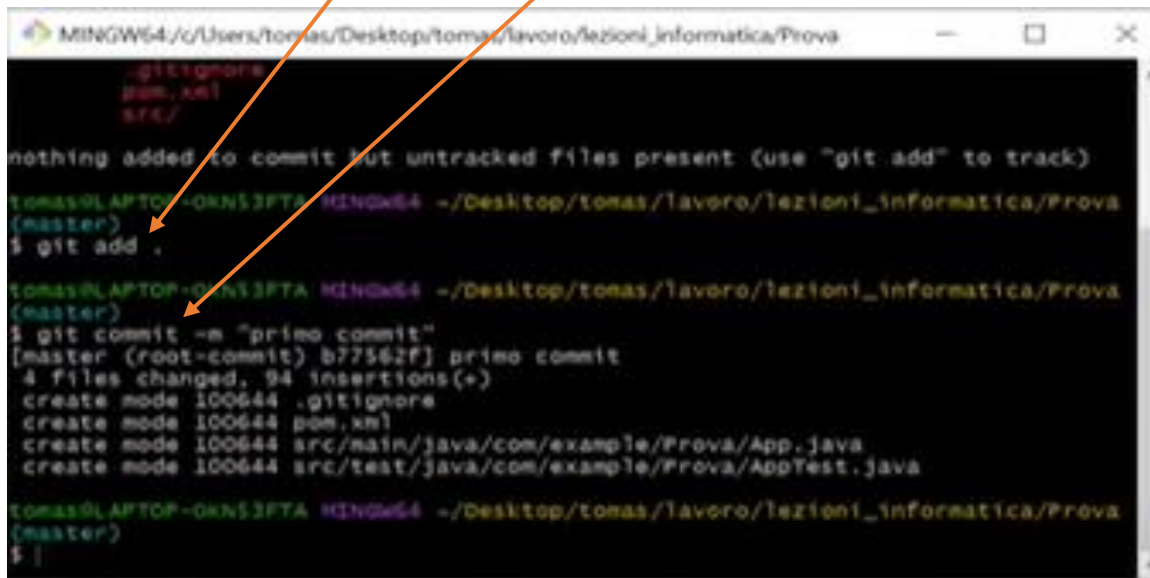
No commits yet

Untracked Files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore
        pom.xml
        src/

nothing added to commit but untracked files present (use "git add" to track)
tomas@LAPTOP-OKN53FTA MINGW64 ~/Desktop/tomas/lavoro/lezioni_informatica/Prova
(master)
$ |
```

7. Successivamente faremo → **git add .** (di tutto).

8. Successivamente faremo il primo → **commit -m "primo commit"** e la schermata sarà pressappoco questa.



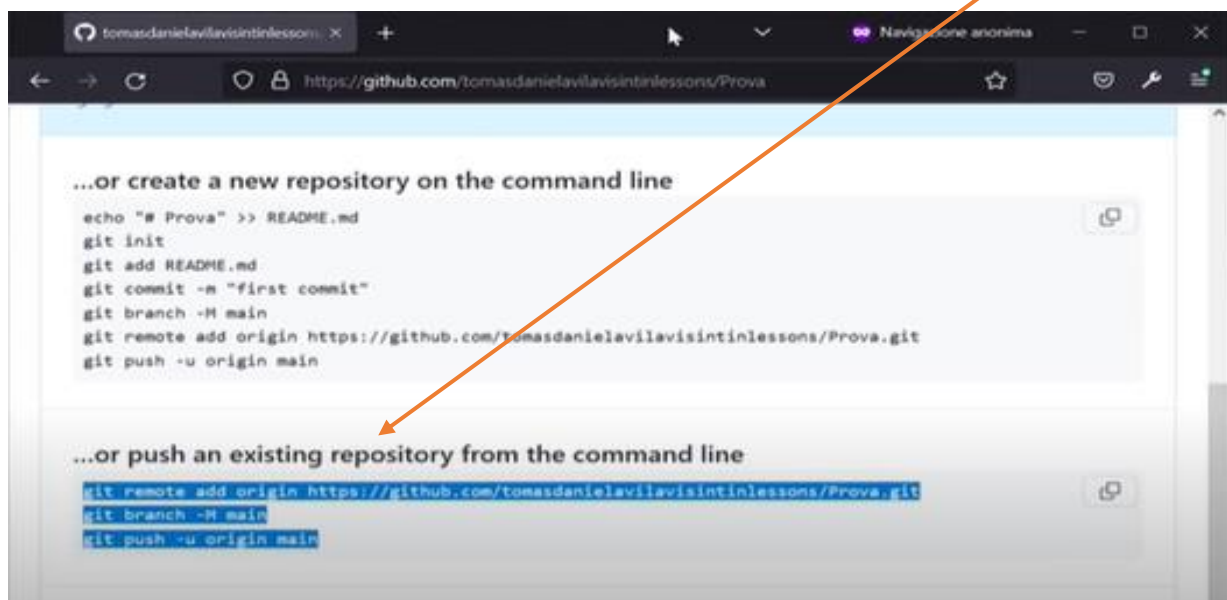
```
MINGW64/c/Users/tomas/Desktop/tomas/lavoro/lezioni_informatica/Prova
$ gitignore
pom.xml
src/

nothing added to commit but untracked files present (use "git add" to track)
tomas@LAPTOP-OKN53FTA MINGW64 ~/Desktop/tomas/lavoro/lezioni_informatica/Prova
(master)
$ git add .

tomas@LAPTOP-OKN53FTA MINGW64 ~/Desktop/tomas/lavoro/lezioni_informatica/Prova
(master)
$ git commit -m "primo commit"
[master (root=commit) b77562f] primo commit
4 files changed, 94 insertions(+)
create mode 100644 .gitignore
create mode 100644 pom.xml
create mode 100644 src/main/java/com/example/Prova/App.java
create mode 100644 src/test/java/com/example/Prova/AppTest.java

tomas@LAPTOP-OKN53FTA MINGW64 ~/Desktop/tomas/lavoro/lezioni_informatica/Prova
(master)
$ |
```

9. Successivamente torneremo su git e nel repository creato troveremo i **comandi** suggeriti per collegarlo in remoto.



```
...or create a new repository on the command line

echo "# Prova" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/tomasdanielavilavisintinlessons/Prova.git
git push -u origin main

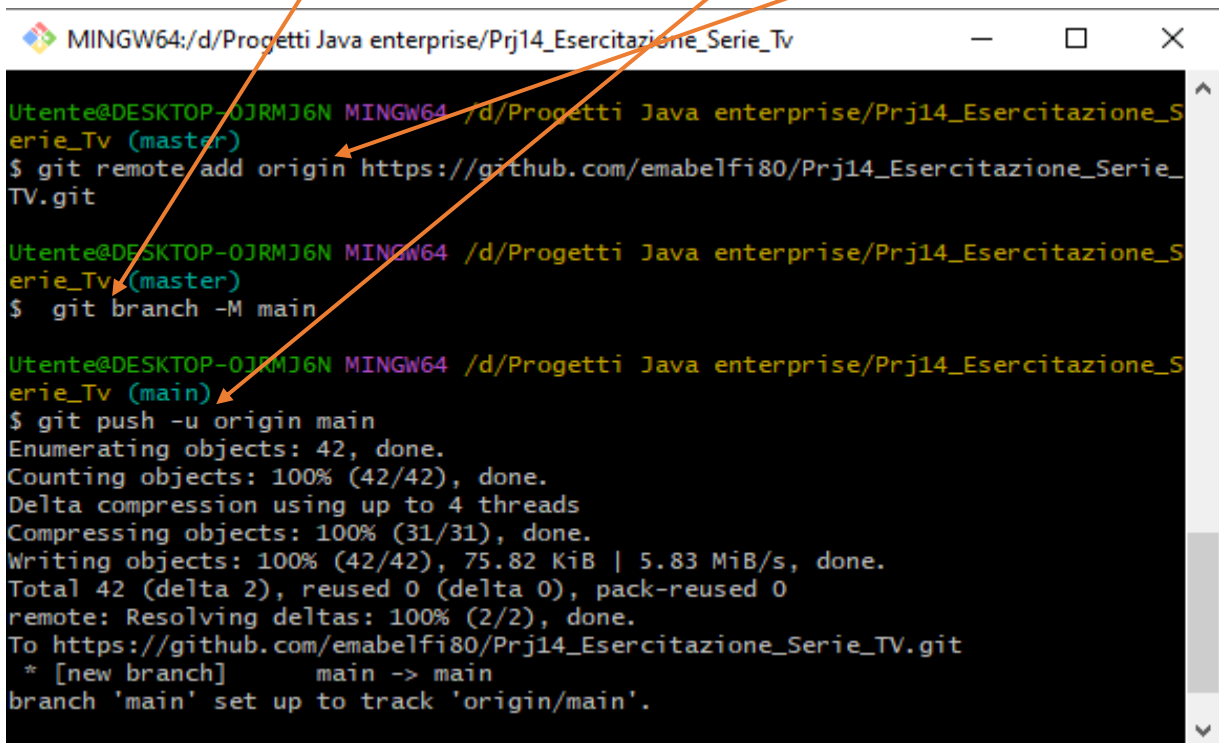
...or push an existing repository from the command line

git remote add origin https://github.com/tomasdanielavilavisintinlessons/Prova.git
git branch -M main
git push -u origin main
```

Naturalmente ci saranno altri comandi utilizzabili eccoli in ordine:

- 1)o creare un nuovo repository dalla riga di comando.
- 2)o eseguire il push di un repository esistente dalla riga di comando. (lo usiamo noi).
- 3)o importare il codice da un altro repository.

10. Successivamente incolleremo in sequenza i comandi → il primo sarà l'**add origin remoto** → il secondo crea il **branch main** → e il terzo eseguirà il **push**.



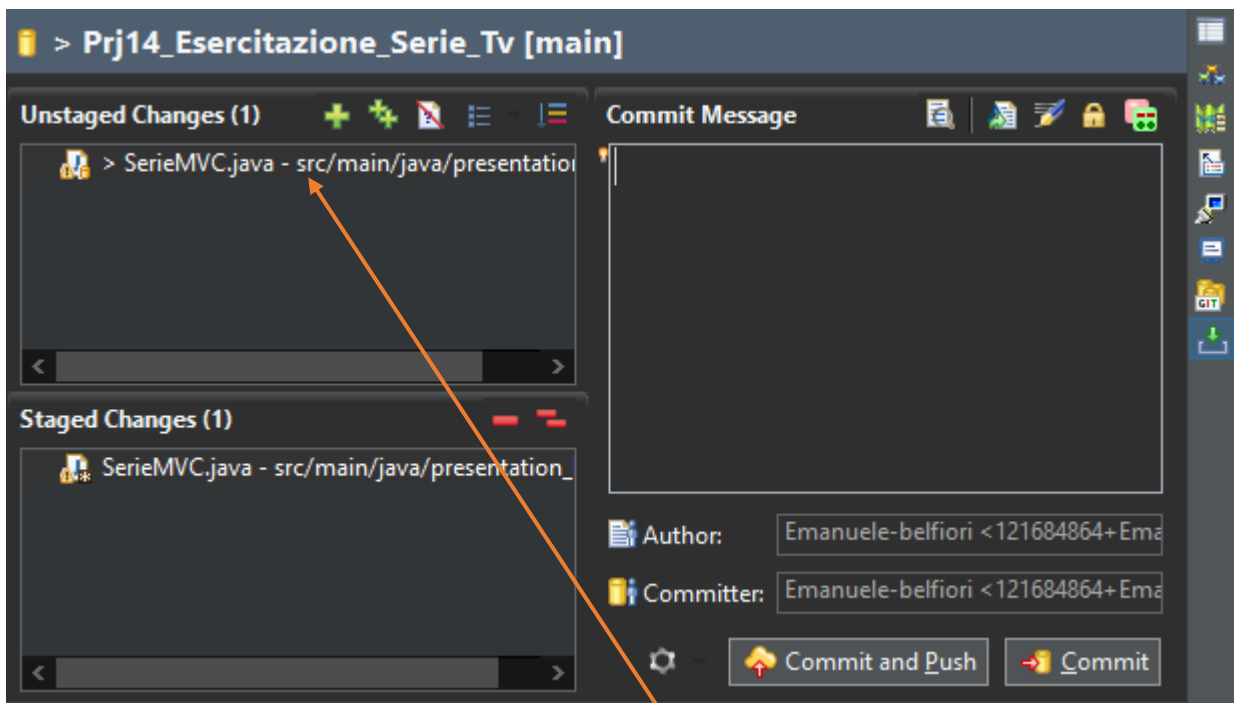
```
MINGW64:/d/Progetti Java enterprise/Prj14_Esercitazione_Serie_Tv

Utente@DESKTOP-0JRMJ6N MINGW64 /d/Progetti Java enterprise/Prj14_Esercitazione_Serie_Tv (master)
$ git remote add origin https://github.com/emabelfi80/Prj14_Esercitazione_Serie_TV.git

Utente@DESKTOP-0JRMJ6N MINGW64 /d/Progetti Java enterprise/Prj14_Esercitazione_Serie_Tv (master)
$ git branch -M main

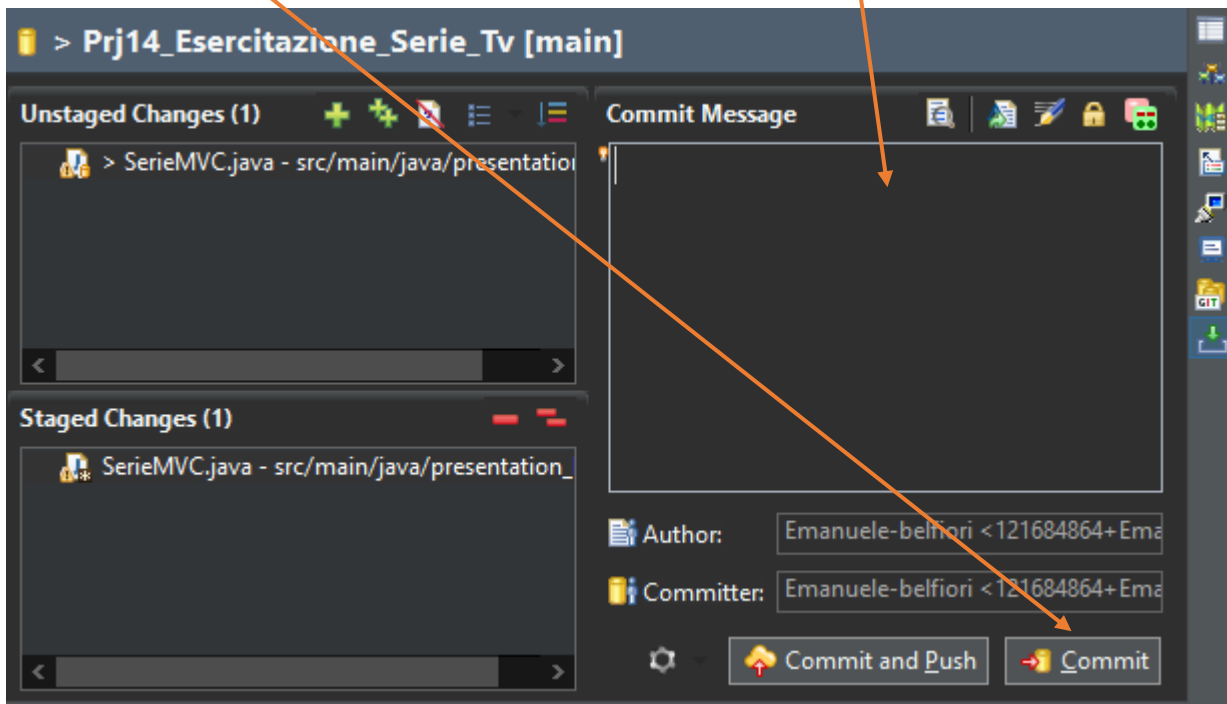
Utente@DESKTOP-0JRMJ6N MINGW64 /d/Progetti Java enterprise/Prj14_Esercitazione_Serie_Tv (main)
$ git push -u origin main
Enumerating objects: 42, done.
Counting objects: 100% (42/42), done.
Delta compression using up to 4 threads
Compressing objects: 100% (31/31), done.
Writing objects: 100% (42/42), 75.82 KiB | 5.83 MiB/s, done.
Total 42 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), done.
To https://github.com/emabelfi80/Prj14_Esercitazione_Serie_TV.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

11. Successivamente torneremo nell'ide apriremo la tendina → windows, poi selezioniamo → Show View poi → Other, poi → cerchiamo **Git Staging**, ci verrà aperta questa finestra.

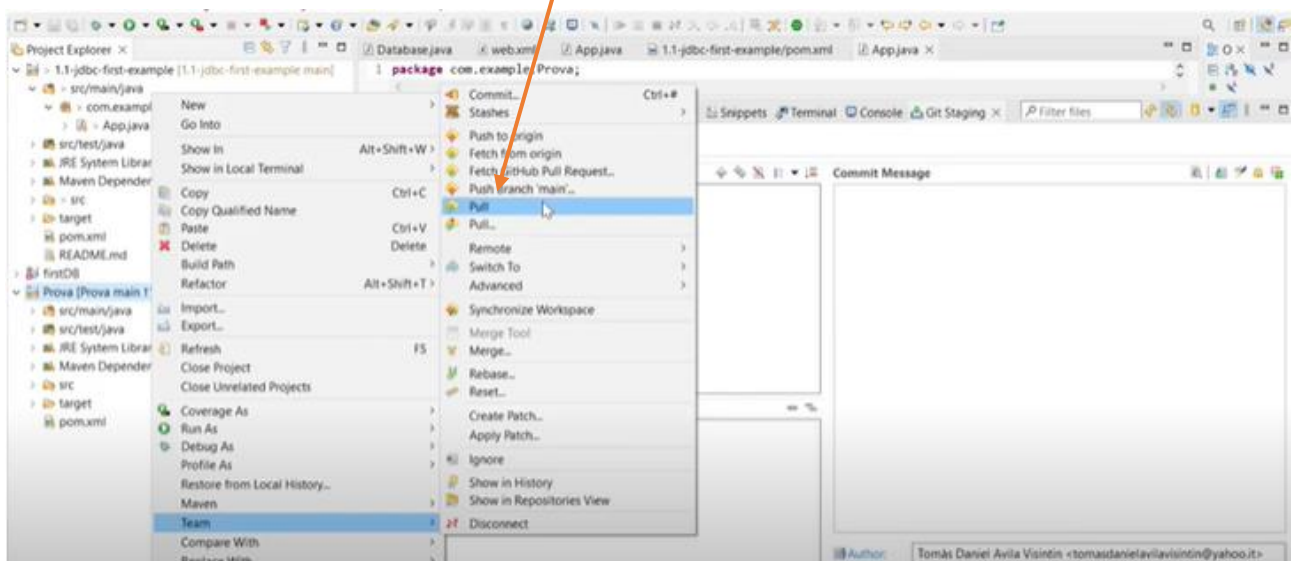


Che ci permette di vedere in tempo reale i cambiamenti che noi apportiamo al progetto, quali e in che cartella del progetto li abbiamo apportati, cliccandoci sopra si aprirà una cartella di consultazione.

12. Se volessimo salvare le modifiche prima scriveremo il **messaggio** “Es: prova commit” poi faremo il **Commit** da qui (notare: così sarebbe esclusivamente nella nostra macchina)

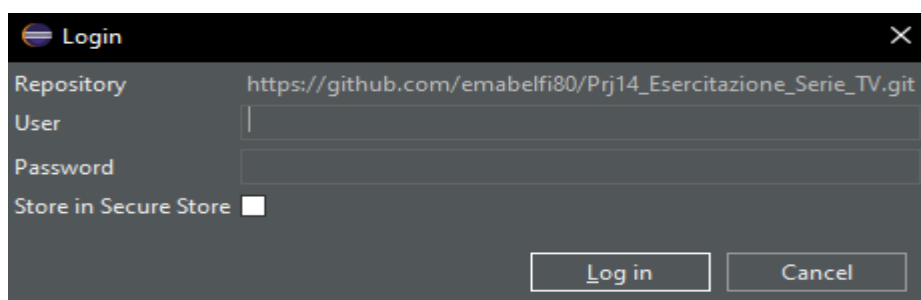


È sempre indicato non fare direttamente Commit and Push, perché prima del Push bisogna sempre fare (soprattutto se si lavora in gruppo) il Pull.

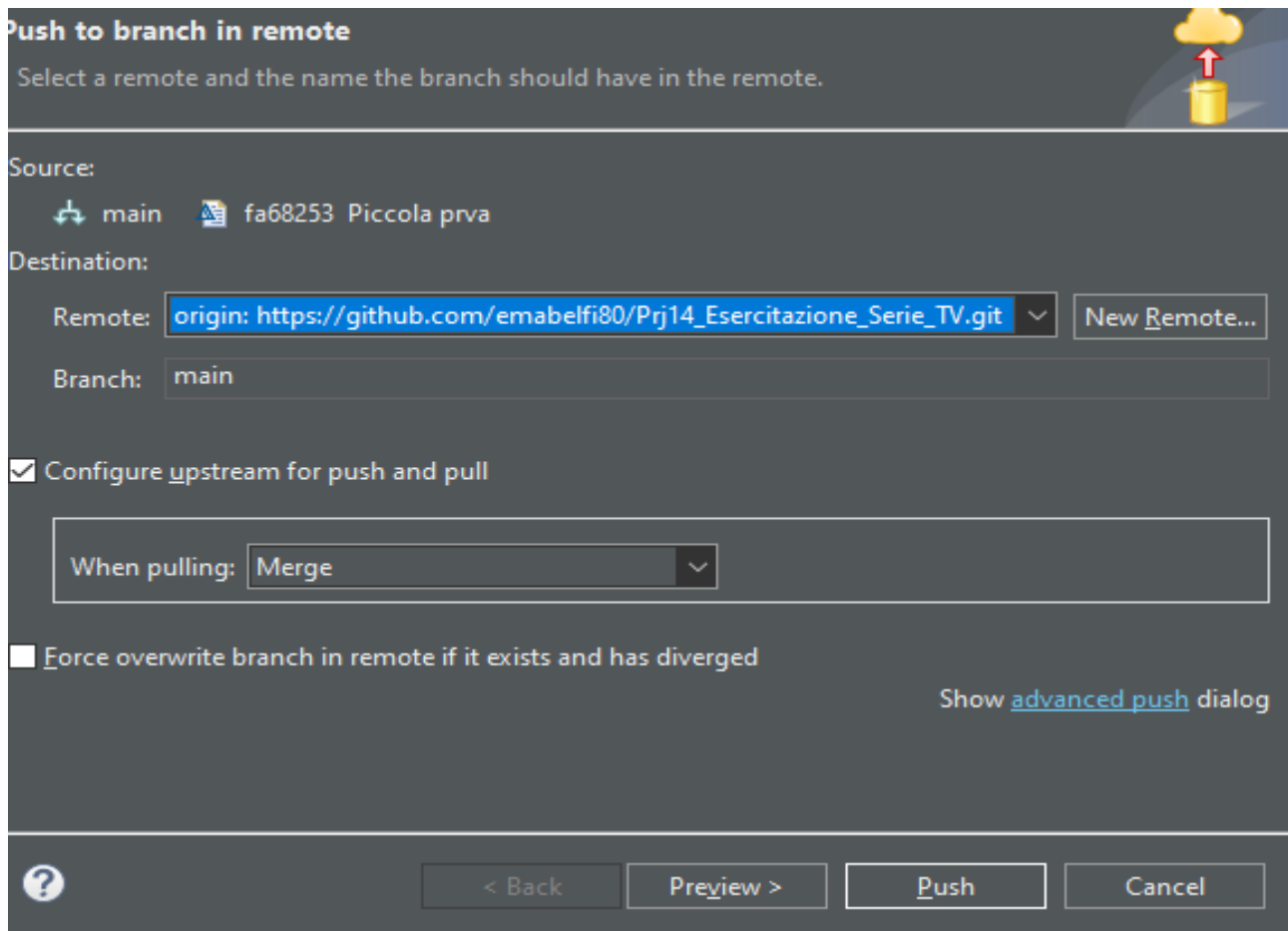


Il comando pull serve per aggiornare la propria repository locale con quella online.

13. Appena fatta questa operazione ci si aprirà questa finestra, ora vi spiego come fare.



14. Dobbiamo andare su git perché non serviranno username e la password classici, dobbiamo creare invece una chiave → **andare in Settings**, poi → **Developer settings**, poi → **Personal acces tokens**, poi → Generate new token (qui puoi inserire i permessi che ti servono e poi verrà generato).
15. Mettere la chiave apposto del campo password spuntare il campo sotto così che non la richieda più → invio, poi fare il → **Push in Git Staging**, poi vi apparirà questa schermata → **confermate il Push ed è finito.**



Push to branch in remote

Select a remote and the name the branch should have in the remote.

Source:

main fa68253 Piccola prva

Destination:

Remote: New Remote...

Branch:

☒ Configure upstream for push and pull

When pulling:

☐ Force overwrite branch in remote if it exists and has diverged

Show [advanced push](#) dialog