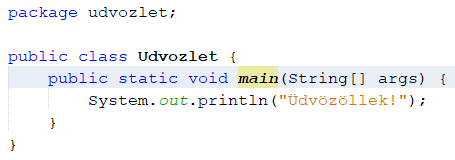
# Git a NetBeans-ben

### Előkészületek

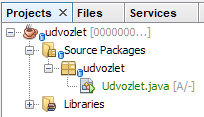
1. Nyisd meg a NetBeans-t, és kezdj egy új Java projektet *udvozlet* néven!
2. Készítsd el az *Udvozlet* osztály első változatát az alábbiak szerint:

* 

1. Futtasd a programot!

### Inicializálás

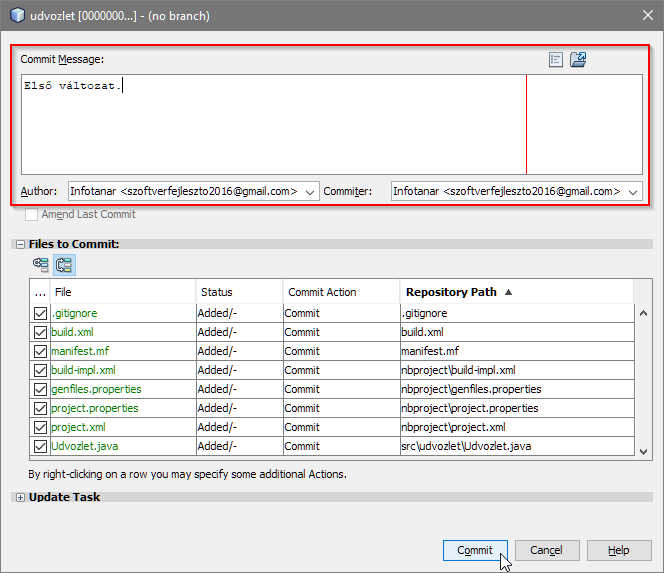
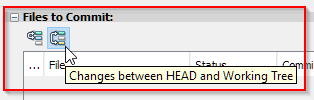
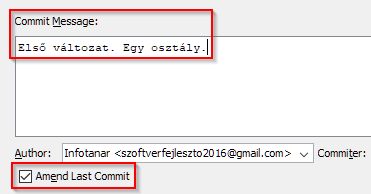
1. Inicializáld a helyi Git repository-t a *Team/Git/Initialize Repository* paranccsal! (Ezután a *Team* menü megváltozik!) A parancs inicializálja a Git repository-t (létrehozza a .git mappát).
2. Nézd meg, hogyan jelöli a NetBeans az új fájlokat, és az azok fölött lévő elemeket a projektben! Kapcsold be a *View/Show Versioning Labels* parancsot, és nézd meg így is!

* 

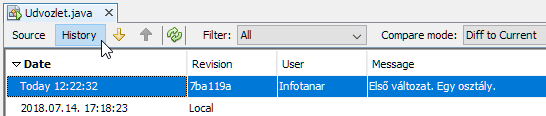
1. Válts a Files fülre, és nézd meg a fájlokat! A NetBeans nem minden fájlt szinkronizál, csak a zölddel jelölteket. A szürkéket ezekből bármikor elő lehet állítani, ezért azok megőrzésére nincs szükség! Ez a beépített verziókezelő használatának legnagyobb előnye.

### Commit

1. A folytatáshoz kapcsold be a Git Repository Browser ablakot (*Team/Repository/Repository Browser*)! Ezután az ablakban a jobb gombbal a repository nevére kattintva megjelenik minden vele kapcsolatos parancs.
2. Válaszd a Commit parancsot, és töltsd ki az alábbiak szerint, majd kattints a Commit gombra! (A saját felhasználónevedet és jelszavadat használd!)

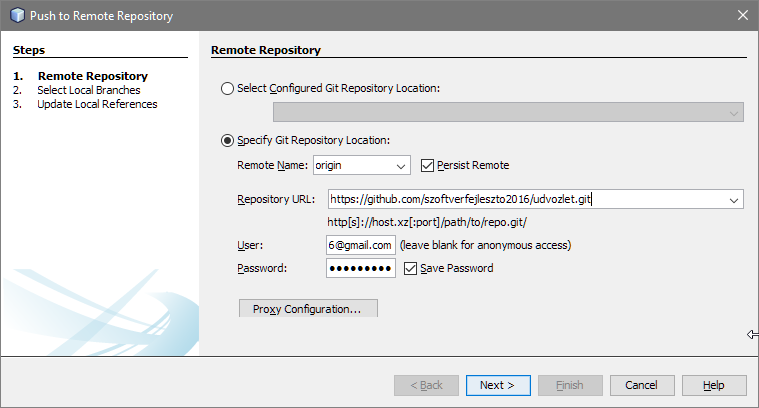
* 
* Az ablak közepén lévő gombok is fontosak:  
  
* Ha a jobb oldali gomb van bekapcsolva (ez az alapértelmezés), akkor minden változást elküld, nem kell előtte a stage-be helyezni őket. A másik gomb esetén csak a stage tartalmát küldi el.
* A commit után figyeld meg a változást a projekt ablakban!
* Előfordulhat, hogy utólag szeretnéd módosítani egy commit szövegét anélkül, hogy új fájlokat küldenél. Ehhez a Commit parancs ablakában az Amend Last Commit kapcsolót kell bekapcsolnod. Próbáld ki:  
  

1. A commitokat a szerkesztő ablak History nézetében tudod megjeleníteni. Próbáld ki!

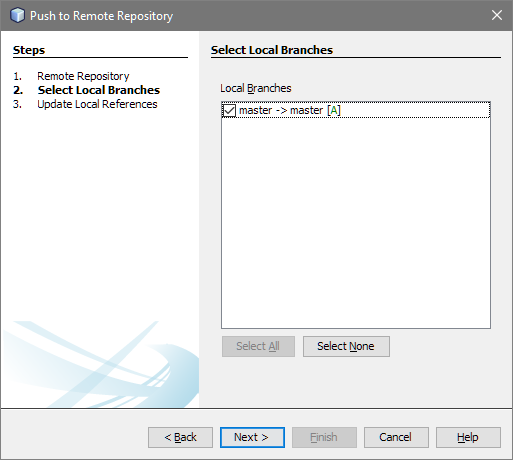
* 

### Távoli repository

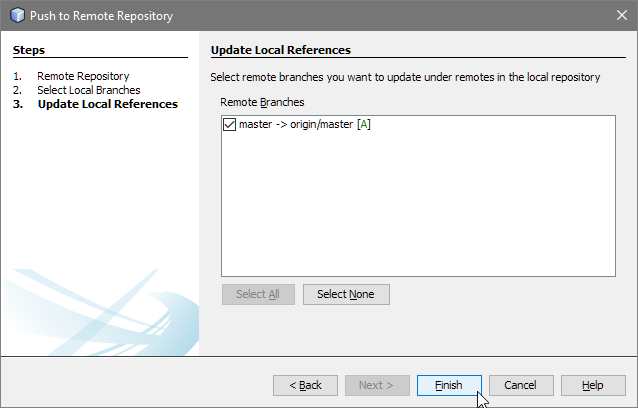
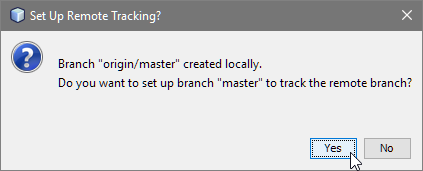
1. Eddig a helyi repository-ba mentettük a fájlokat. Nyisd meg a GitHub oldalad (jelentkezz be), és készíts egy új, üres repository-t *udvozlet* néven!
2. A létrehozás után megjelenő oldalon másold a vágólapra az új repository címét!
3. Válaszd a projekt menüjéből a Remote/Push... parancsot! Az első ablakot töltsd ki az alábbiak szerint, majd kattints a Next gombra!

* 

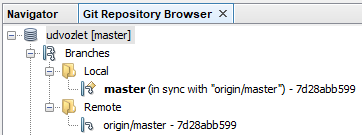
1. Ezután jelöld meg a master ágat, majd kattints ismét a Next-re!

* 

1. A következő ablakban kattints a Finish gombra, majd kattints az OK gombra!

* 
* A következő ablakban kattints a Yes gombra!
* 

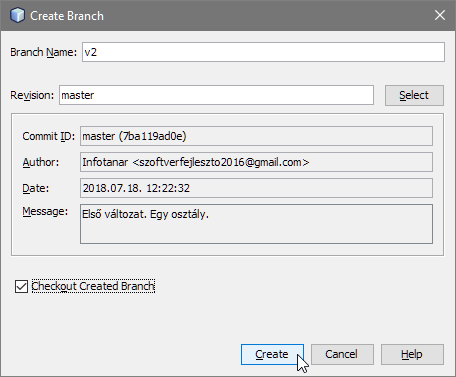
1. Nézd meg a beállított távoli repository-t a Git Repository Browserben!

* 

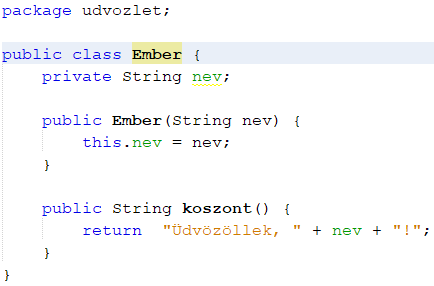
1. Frissítsd a GitHub oldalt, és ellenőrizd a távoli repository-t!

### Elágazás

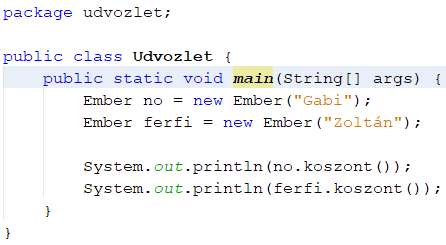
1. Kezdj egy új ágat a helyi repository-ban a Branch/Tag, Create Branch paranccsal, és válts át rá! Így töltsd ki a megjelenő ablakot:

* 

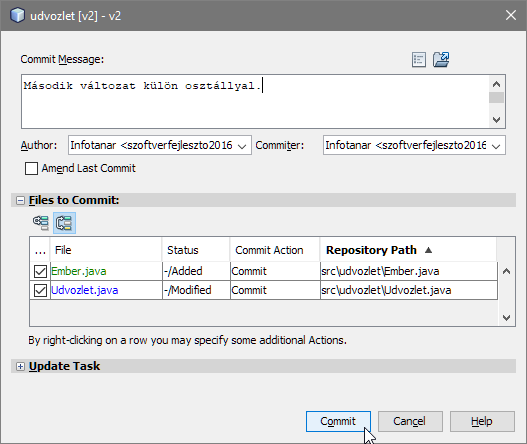
1. Készíts egy új osztályt *Ember* néven az udvozlet csomagban az alábbiak szerint:

* 

1. Módosítsd a főprogramot az *Udvözlet* osztályban így:

* 
* Futtasd a programot!

1. Figyeld meg, hogyan jelöli a NetBeans a változásokat, majd commit-old őket!

* 

1. Válts át a master ágra, majd vissza a zöld ágra! Ehhez kattints duplán az ág nevére a Git Repository Browserben, majd nyomd meg a Checkout gombot! (A Select gombbal az ágon belül a változat is kiválasztható.)
2. A következő lépés a zöld ág feltöltése a távoli repository-ba. Ehhez válaszd a projekt menüjében a Remote/Push... parancsot, majd töltsd ki az ablakokat. A második lépésben a v2 ágat válaszd!
3. Figyeld meg, hogy a Remote repository-ban is megjelent a v2 ág! Ellenőrizd a GitHub-on is!

### Módosítások letöltése

1. Módosítsd a GitHub-on az Udvozlet.java fájlt úgy, hogy Gabi helyett Kati szerepeljen benne! Commit-old a módosítást "Gabi helyett Kati." szöveggel a v2 ágba! (Ezt a csapat másik tagja is módosíthatta volna.)
2. Töltsd le a módosításokat a Remote/Pull… paranccsal a v2 ágba, és ellenőrizd a változásokat!

### Ágak egyesítése

1. Helyben úgy egyesíthetnéd az ágakat, hogy a átváltasz a master ágra, a jobb gombbal rákattintasz a v2 ágra, majd kiválasztod a *Merge revision...* parancsot.   
   Most azonban a GitHub-on szeretnénk az egyesítést elvégezni, ezért válts a GitHub oldaladra, és az udvozlet projektben válaszd a *Pull request* fület!
2. Kattints a *New pull request* gombra, válaszd ki a v2 ágat, majd kattints a *Create pull request* gombra!
3. Írj egy üzenetet, majd itt is kattints a *Create pull request* gombra!
4. Erősítsd meg a műveletet a *Merge pull request*, majd a *Confirm merge* gomb megnyomásával!
5. Végül töröld a zöld ágat a GitHub-ról a *Delete branch* gombbal!
6. Válts a NetBeans-re, válts vissza a master ágra, és töltsd le a master ág módosításait! (Jobb kattintás a helyi master ágra, Sync with origin/master)
7. Töröld a Local és a Remote zöld ágakat is! (jobb kattintás, Delete Branch)
8. A GitHub-ról is töröld a v2 ágat!

### Klónozás

Ha egy másik számítógépen szeretnéd folytatni a projektet, akkor klónozni kell arra a gépre a GitHub-ról.

Ezt most úgy modellezzük, hogy egy másik mappába klónozzuk a projektet a távoli repository-ból.

1. Zárd be a projektet a NetBeans-ben, és válaszd a Team/Git/Clone parancsot!
2. Add meg a repository URL címét, és célnak adj meg egy új mappát! Nyisd meg a projektet!
3. Figyeld meg, hogy a NetBeans jelzi, hogy az osztályokat újra le kell fordítani! Futtasd a programot!