# Download project locally from Azure DevOps

* Azure DevOps -> Repos -> Files -> Clone
* Copy HTTPS
* in local folder of the entire project -> git bash here
* git clone <URL> -> will create BE/ FE

# Azure DevOps - create and start a new task

* Backlogs
  + new work item
  + title (ex: AddUserRoles)
  + add description
  + view as board
  + assign to member
  + upload status

* Git Extensions
  + select project
  + update local version
    - right click on master -> checkout
    - fetch and prune all
    - pull merge
  + create branch
    - right click on master -> create branch
    - naming convention: "2-AddUserRoles"

# Pull request, merge, rebase

* Git Extensions (do this for every significant change)
  + do work on the branch
  + check all unit tests
  + commit
  + stage only the needed files, reset the others (for a migration, keep all three files: .cs, .Designer.cs) (also keep the .csproj file) (can also reset some changes by selecting them and pressing R)
  + write commit message
  + commit and push (yes to new for remote and to add tracking reference)

* Azure DevOps (after finishing the task)
  + check all unit tests
  + check files
  + pull request
  + create pull request (or new pull request if I want for another branch)
    - add title ex: "2 Add User Roles"
    - add mentors and colleagues as reviewers
    - add work item
    - Create
  + Write comment on Teams channel (with link)
  + colleagues comment and approve (maybe you need to make modifications -> then reply with "fixed" and resolve)
    - I think that you cannot complete PR if there are unresolved comments
  + Mentors comment and approve

## Method 1 (for the first PR, or when you master is not behind – it was not updated since your last PR):

* Azure DevOps
  + pull request
  + COMPLETE
    - squash commit
    - Customize merge commit message "Merged 2 Add User Roles"
    - Complete merge

* Git Extensions
  + update master branch
    - right click on master -> checkout
    - pull-merge + fetch and prune all
    - delete local branch

## Method 2 (for the other PR):

* Git Extensions
  + update master branch
    - right click on master -> checkout
    - pull-merge + fetch and prune all
  + checkout working branch
    - right click on working branch -> checkout
  + rebase branch
    - right click on master -> rebase (first option)
  + solve conflicts and push force with lease
    - solve conflicts
    - for each file open in kdiff3
    - save
    - continue rebase
    - revert the commit with .orig file
    - check in code
    - push
    - show options -> FORCE PUSH WITH LEASE
    - push

* Azure DevOps
  + pull request
  + COMPLETE
    - squash commit
    - Customize merge commit message "2 Add User Roles"

* Git Extensions
  + right click on master -> checkout
  + fetch and prune all
  + pull merge
  + delete old branch (which is now gone)