

Basic GIT

- Simple tutorial -

GIT

=

Distributed version control system

Git repository

- The history of a collection of files starting from a certain directory
 - Why is such a thing useful?
- **GitHub** : a web-based git repository hosting
 - + issue tracker, wiki support, code review, pull requests support, etc.

Let's start!

- Create an account at <https://github.com>

A screenshot of the GitHub sign-up form. It features three white input fields on a dark background. The first field is labeled 'Pick a username', the second 'Your email address', and the third 'Create a password'. Below the password field, there is a line of text: 'Use at least one letter, one numeral, and seven characters.' A prominent green button with the text 'Sign up for GitHub' is positioned below the text. At the bottom, a line of smaller text states: 'By clicking "Sign up for GitHub", you agree to our terms of service and privacy policy. We'll occasionally send you account related emails.'

Crearea unui repository

The screenshot shows the GitHub profile page for a user named 'ipuser'. The user's profile picture is a green and white pixelated pattern. The page header includes the GitHub logo, a search bar, and navigation links for 'Pull requests', 'Issues', and 'Gist'. A dropdown menu is open, displaying four options: 'New repository', 'Import repository', 'New gist', and 'New organization'. The user's statistics are shown as 'Overview', 'Repositories 0', 'Stars 0', 'Followers 0', and 'Following 0'. The 'Popular repositories' section contains a message: 'You don't have any public repositories yet.' At the bottom, it states '1 contribution in the last year' and provides a link to 'Contribution settings'.

Search GitHub

Pull requests Issues Gist

ProTip! Updating your profile with your name, location, and a profile picture helps GitHub users get to know you.

New repository
Import repository
New gist
New organization

Overview Repositories 0 Stars 0 Followers 0 Following 0

Popular repositories

You don't have any public repositories yet.

1 contribution in the last year Contribution settings

ipuser


Add a bio

Edit profile

Crearea unui repository

Owner

Repository name

 ipuser ▾ / mycoolproject 

Great repository names are short and memorable. Need inspiration? How about **crispy-palm-tree**.

Description (optional)

Simple demo project

-
- ☒  **Public**
Anyone can see this repository. You choose who can commit.
- ☐  **Private**
You choose who can see and commit to this repository.

☐ **Initialize this repository with a README**

This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** ▾

Add a license: **None** ▾



Create repository

Creare cu succes

Quick setup — if you've done this kind of thing before

 Set up in Desktop or **HTTPS** **SSH** <https://github.com/ipuser/mycoolproject.git>



We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

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

```
echo "# mycoolproject" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/ipuser/mycoolproject.git
git push -u origin master
```



Clonarea pe masina locala

- Instalare distribuție git: <https://git-scm.com/downloads>
- Windows: <https://git-scm.com/download/win>
- In linia de comanda:

```
git clone https://github.com/ipuser/mycoolproject.git
```

- Branch: master vs. origin

Studiu de caz: adaugarea unui README

1. Creare fisier/fisiere

1. Se creeaza fisierul README.md (in folderul local mycoolproject)
2. Se adauga urmatorul continut:

```
## My cool project
```

```
-----
```

```
This is a simple demo.
```

2. Pregatirea pentru commit :

```
git add README.md
```

3. Commit local

```
git commit -m "Added README file."
```

4. Trimiterea modificarilor:

```
git push origin master
```

Pe Github:



ipuser Added README file.

Latest commit 8e780ab 33 seconds ago



README.md

Added README file.

32 seconds ago



README.md

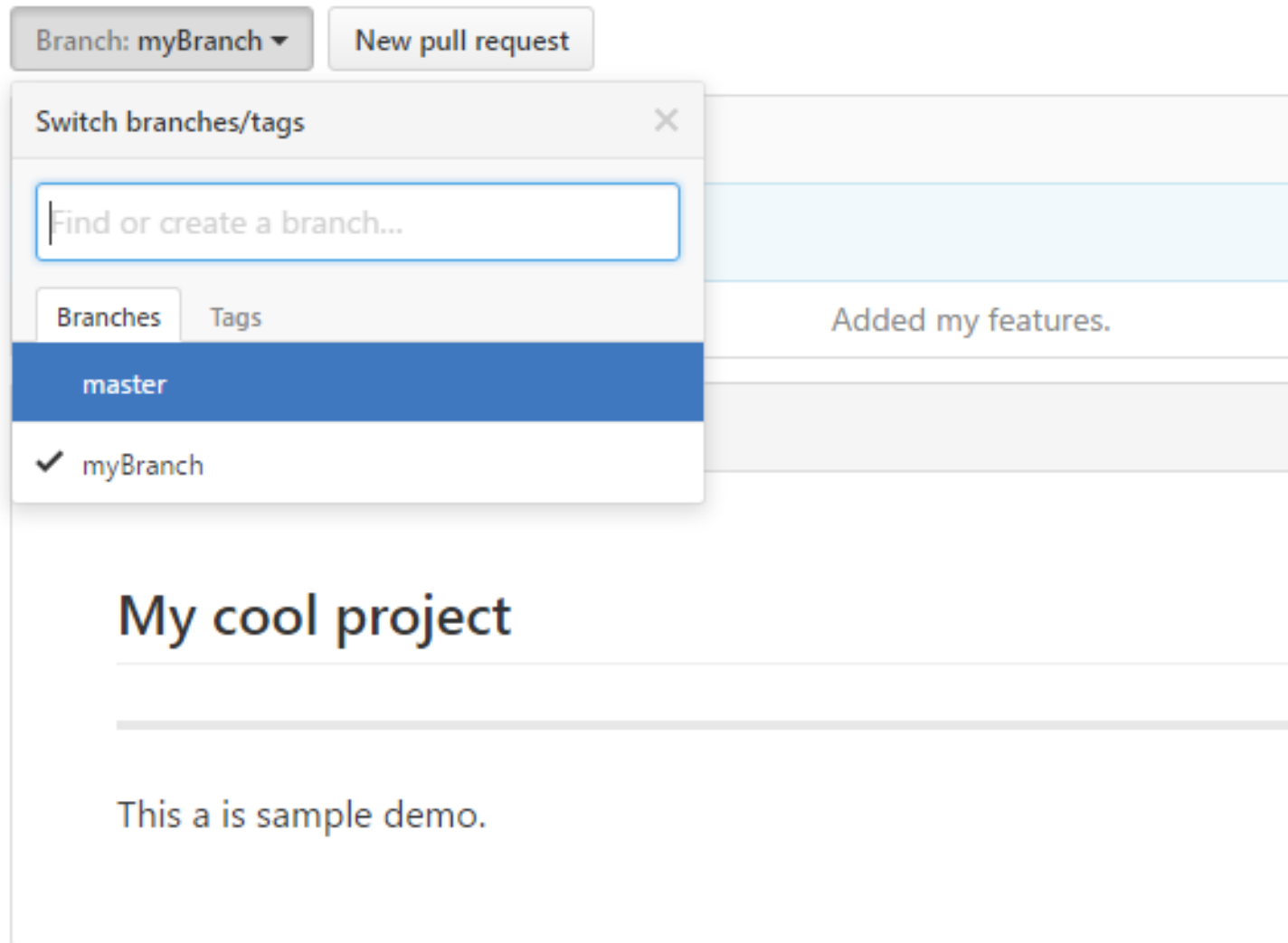
My cool project

This is simple demo.

Branch-uri

- De cele mai multe este bine sa NU lucram direct in master
- Este preferabil un branch (o copie) in care putem lucra independent de ceilalti
- Pentru a crea un branch:
`git checkout -b myBranch master`
- Pentru a face push unui branch:
`git push origin myBranch`

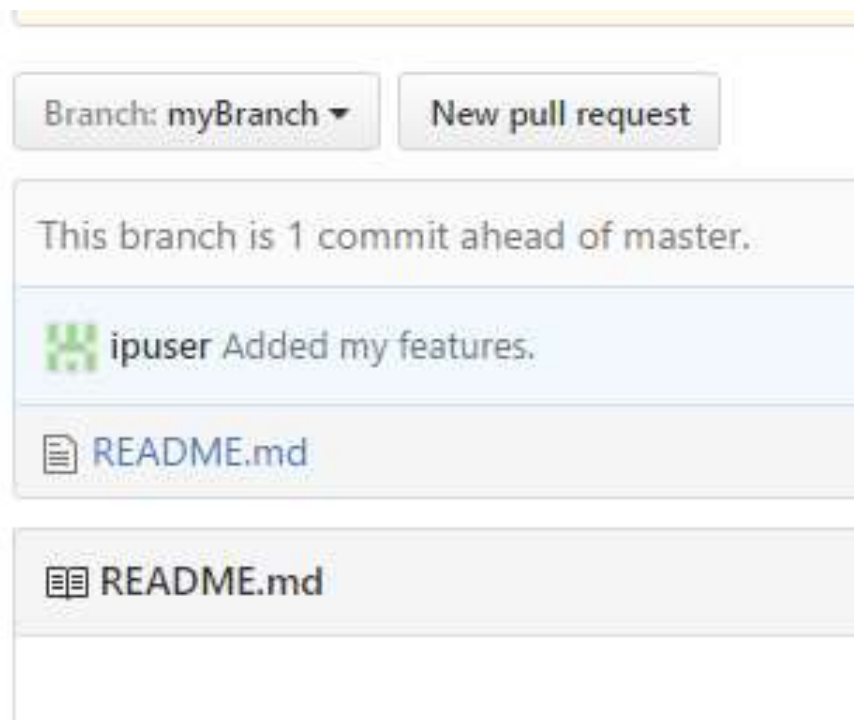
Branch-ul este vizibil pe GitHub



Pull request

1. Cum ajung modificarile in master?

Apasati “New pull request”



Added my features. #1



ipuser wants to merge 1 commit into master from myBranch



Conversation 0



Commits 1



Files changed 1



ipuser commented just now

Owner



Modified README.



Added my features.

ce203b1

Add more commits by pushing to the myBranch branch on ipuser/mycoolproject.



This branch has no conflicts with the base branch

Merging can be performed automatically.

Merge pull request



You can also [open this in GitHub Desktop](#) or [view command line instructions](#).



Write

Preview

AA B i



Leave a comment

Code review



ipuser commented 4 minutes ago

Owner



Modified README.



Added my features.

ce203b1




Code review

Changes from all commits ▾ 1 file ▾ +1 -1   

Added my features.



ipuser committed 15 minutes ago

2    README.md

... ... @@ -1,4 +1,4 @@

1 1 ## My cool project

2 2

3 3 -----

4 -This is simple demo.

4 +This a is sample demo.

Unified

Split

Review changes ▾

Submit your review

Review summary

Leave a comment

☒ Comment

Submit general feedback without explicit approval.

☐ Approve

Submit feedback and approve merging these changes.

☐ Request changes

Submit feedback that must be addressed before merging.

Submit review

Merging



This branch has no conflicts with the base branch

Merging can be performed automatically.

Merge pull request



You can also [open this in GitHub Desktop](#) or [view command line instructions](#).



Merge pull request #1 from ipuser/myBranch

Added my features.


Confirm merge

Cancel



Pull request successfully merged and closed

You're all set—the `myBranch` branch can be safely deleted.

 **Delete branch**

Concluzii

- GIT: “versioning system” distribuit
- Fiecare lucreaza pe copia proprie
- Schimbarile sunt vizibile abia la pull request
- Se face code review -> calitate crescuta a codului
- Nu se altereaza masterul, ci se adauga doar modificarile testate si verificate din branch-uri