

Git & GitHub

QUICK INTRODUCTION

Introduction to Git as a version control system:
concepts, main features and practical aspects.

Luigi De Russis and Fulvio Corno



Politecnico
di Torino



Goal

- What is Revision Control?
- What is Git?
- What is GitHub?
- How to access Revision Control with Git and GitHub from within Eclipse?
- What are the Eclipse workflows useful in this course?

Version Control Systems

Record changes to a file or a set of files over time so that you can recall specific versions later

Three generations:

1. Local (RCS, SCCS)
2. Centralized (CVS, Subversion, Team Foundation Server)
3. Distributed (Git, Mercurial)



Basic Concepts

Repository

- place where you store all your work
- contains every version of your work that has ever existed
 - files
 - directories layout
 - history
- can be shared with the whole team



REPOSITORY

Basic Concepts

Working copy

- a snapshot of the repository used for... working
- the place where changes happens
- private, not shared with the team
- it also contains some metadata so that it can keep track of the state of things
 - has a file been modified?
 - is this file new?
 - has a file been deleted?

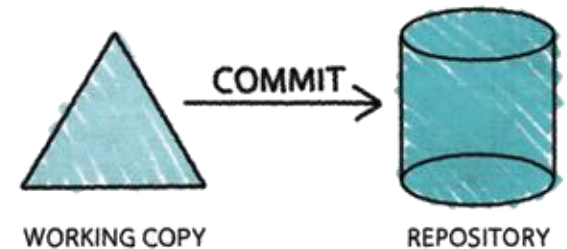


WORKING COPY

Basic Concepts

Commit

- the operation that modifies the repository
- atomically performed by modern version control tools
 - the integrity of the repository is ensured
- it is typical to provide a log message (or comment) when you commit
 - to explain the changes you have made
 - the message becomes part of the history of the repository



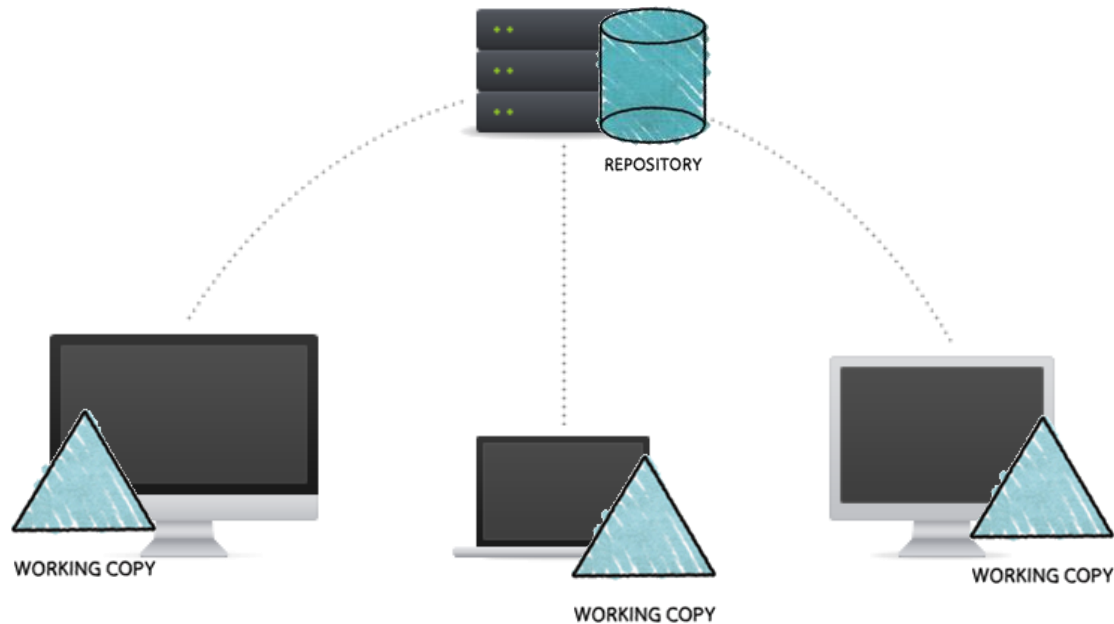
Basic Concepts

Update

- update the working copy with respect to the repository
 - apply changes from the repository
 - merge such changes with the ones you have made to your working copy, if necessary

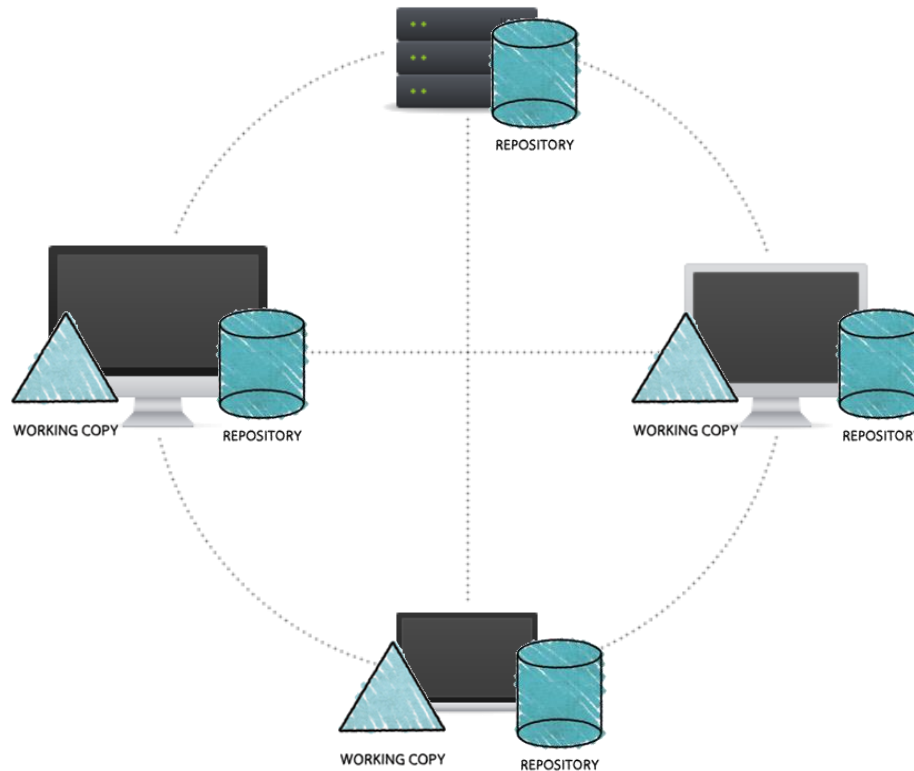


Centralized Version Control



- one central repository
- client-server relationship

Distributed Version Control

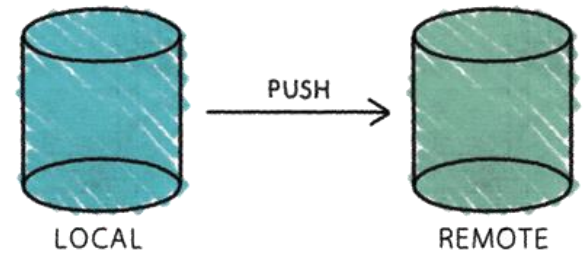


- clients and server have the full copy of the repository
 - local repositories 'clone' a remote repository
- it is possible to have more than one server

More Basic Concepts

Push

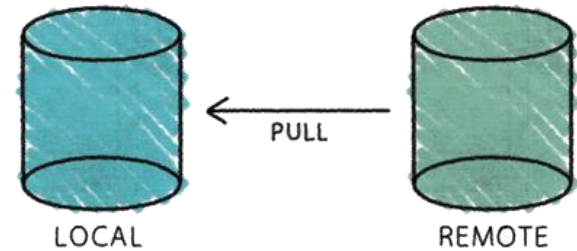
- copy changesets from a local repository instance to a remote one
 - synchronization between two repository instances



More Basic Concepts

Pull

- copy changesets from a remote repository instance to a local one
 - synchronization between two repository instances



Introducing... Git



- Distributed Version Control System
- Born
 - on 2005 for the Linux kernel project
 - to be used via command line
- Website: <http://git-scm.com>
- Highlights:
 - free and open source
 - strong support for non-linear development
 - fully distributed
 - efficient handling of large projects
 - cryptographic authentication of history

Getting started with Git

- Standard installations
 - <http://git-scm.com/downloads>
- Available for all the platform
- Git Graphical Applications
 - <http://git-scm.com/downloads/guis>
 - Suggestion: GitExtensions, SourceTree
- **For this course, Git is**
 - **integrated in Eclipse (plugin “EGit”)**

Hosted Git

- To have (at least) one remote repository
 - alternative: set up your own Git server!
- Most popular:
 - **GitHub**, <https://github.com/>
 - Bitbucket, <https://bitbucket.org/>
 - GitLab, <https://about.gitlab.com/gitlab-com/>

GitHub



- Slightly different than other code-hosting sites
 - instead of being primarily based on the project, it is user-centric
 - social coding
- Owned by Microsoft
 - free account to host as many open source project as you want
 - free plans for students
 - <https://education.github.com>

For Labs

- Create a personal GitHub account
 - You will have “education” discounts if you use your University e-mail
 - <https://education.github.com>
- Try Git!
 - <http://try.github.io/>
 - 15 minutes tutorial

Password vs Token

- Your GitHub account is associated with a password, that is **valid for interactive** usage, only
- The same password can not be used by Eclipse when it tries to push your modifications
- Eclipse needs to use a **Personal Access Token** to access
 - Create on GitHub
 - Save it on your files
 - Store inside Eclipse

Creating a Token (1/2)

The image shows a sequence of steps to create a GitHub token. It starts with a user profile dropdown menu where 'Settings' is selected. This leads to the 'Account' settings page, where 'Developer settings' is chosen. From there, 'Personal access tokens' is selected in the left sidebar. The final view shows the 'Personal access tokens' page with a 'Generate new token' button highlighted.

Signed in as fulcornio

- Set status
- Your profile
- Your repositories
- Your codespaces
- Your organizations
- Your projects
- Your stars
- Your gists
- Upgrade
- Feature preview
- Help
- Settings**
- Sign out

ACCOUNT

- Appearance
- Account security
- Billing & plans
- Security log
- Security & analysis
- Emails
- Notifications
- Scheduled reminders
- SSH and GPG keys
- Repositories
- Packages
- Codespaces
- Organizations
- Saved replies
- Applications
- Developer settings
- Moderation settings

Personal access tokens

Generate new token Revoke all

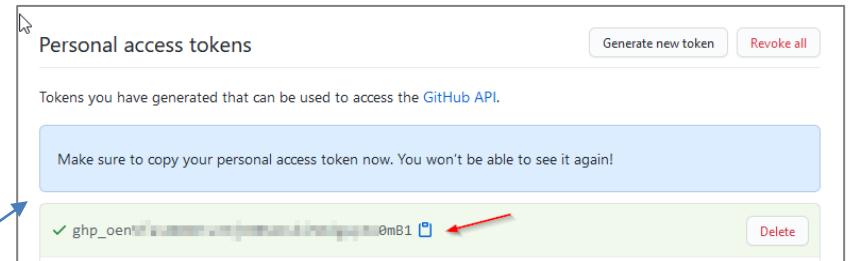
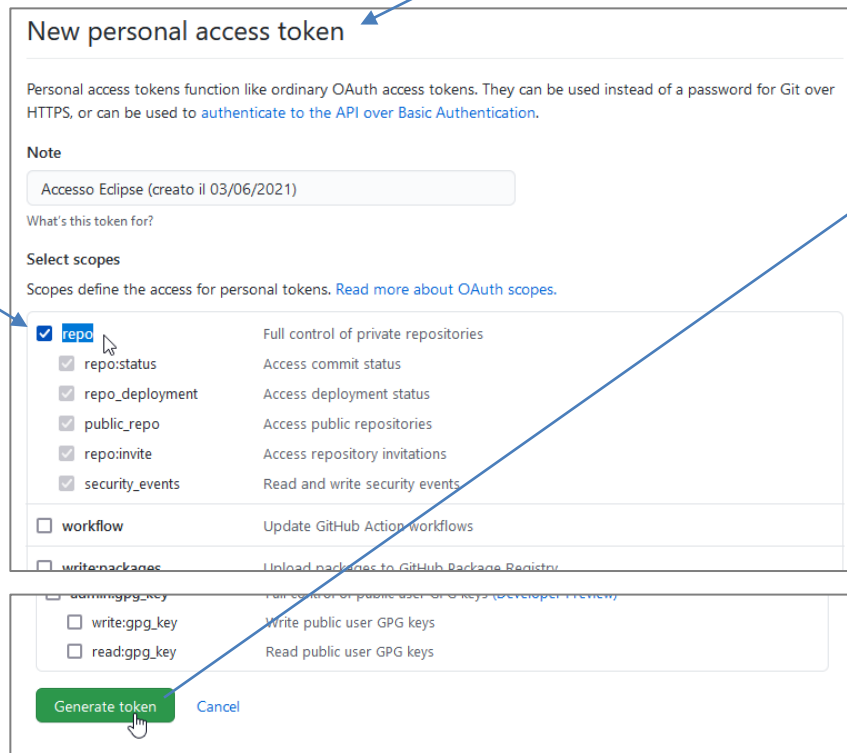
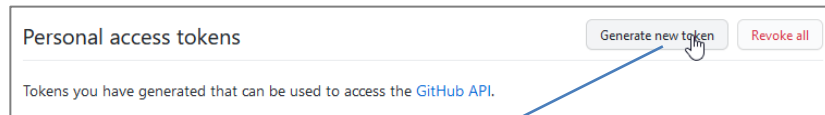
Tokens you have generated that can be used to access the GitHub API.

GitHub Apps

OAuth Apps

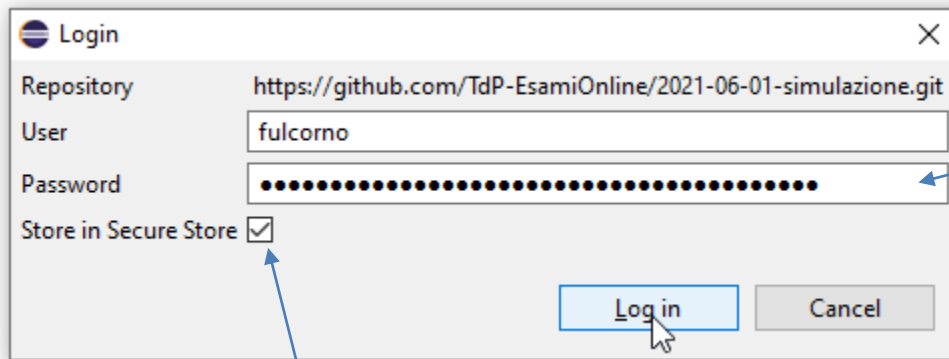
Personal access tokens

Creating a Token (2/2)



- Check the expiration date (no expiration)
- Copy by clicking on icon (not with ^C)
- Paste/save into Notepad or similar application (not Word, not screenshot...)
 - Don't lose this file

Inserting a Token into Eclipse



Insert token here

Check this, to avoid being asked 1000 times

Workflow 1: “Create new project”

1. Create a project in Eclipse (normally, or using Maven Archetypes)
2. Create the local repository in Eclipse (**Team|Share**)
3. Create a new (empty) project in GitHub
4. Push changes (**Team|Commit&push**)

New Project on GitHub

Repositories 6 + New repository


Find a repository...

All Forks Sources Private Public

- TdP-2016/**EsIndovinaNumero**
- TdP-2016/**Lab1**
- TdP-2016/**Lab0**
- TdP-2016/**EsContaLettere**
- TdP-2016/**WordSet_dumb**
- TdP-2016/**materiale**

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner:  TdP-2016 / Repository name:

Great repository names are short and memorable. Need inspiration? How about curly-pancake.

Description (optional)

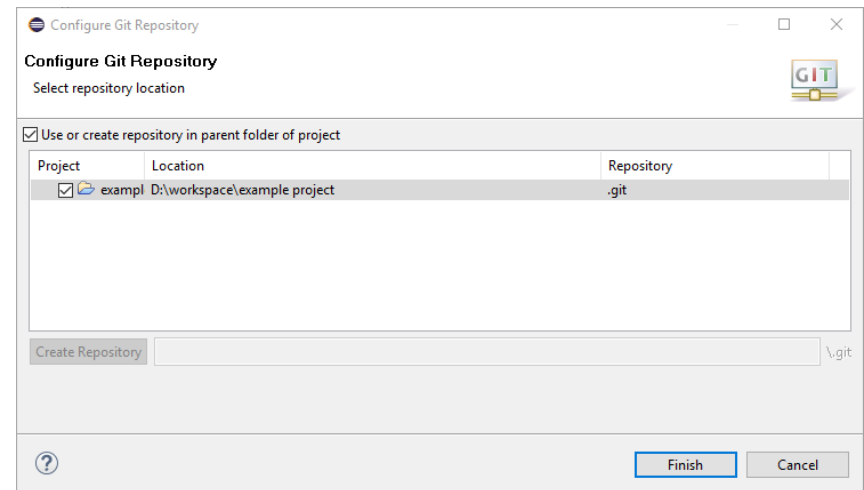
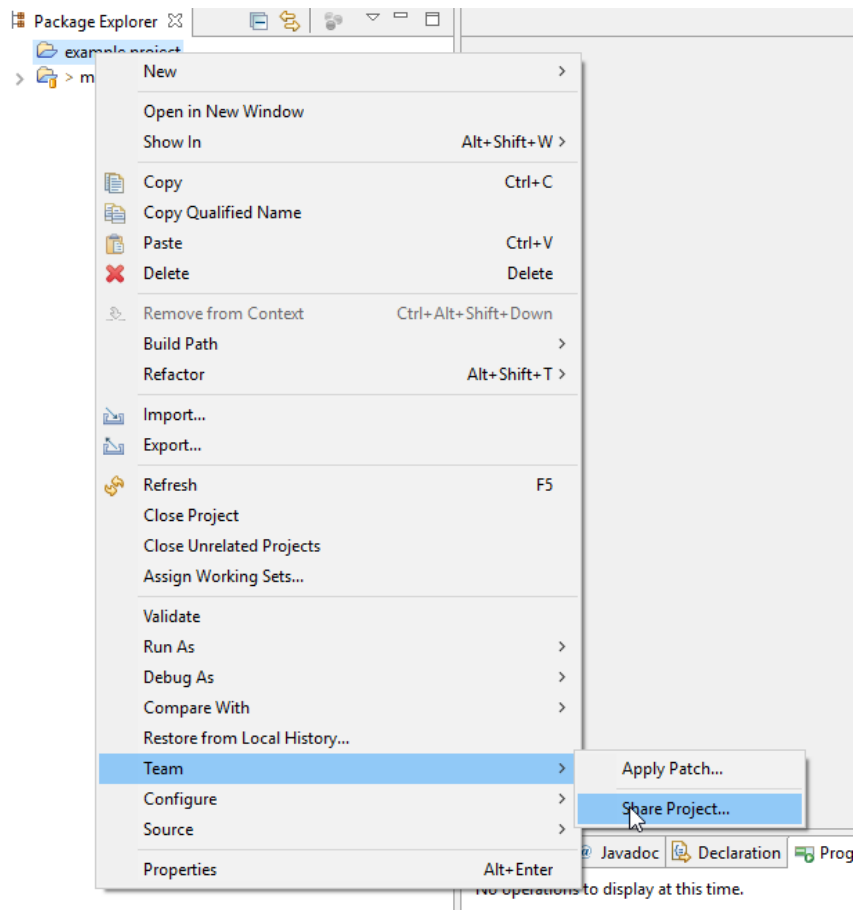
- Public**
Anyone can see this repository. You choose who can commit.
- Private**
You choose who can see and commit to this repository.

- 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.

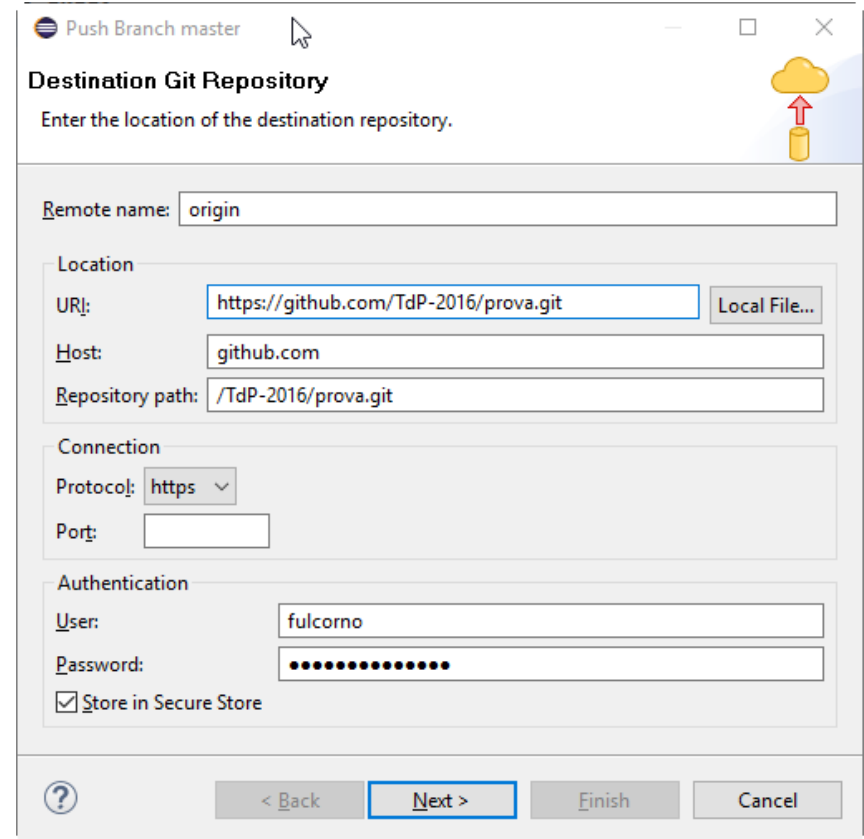
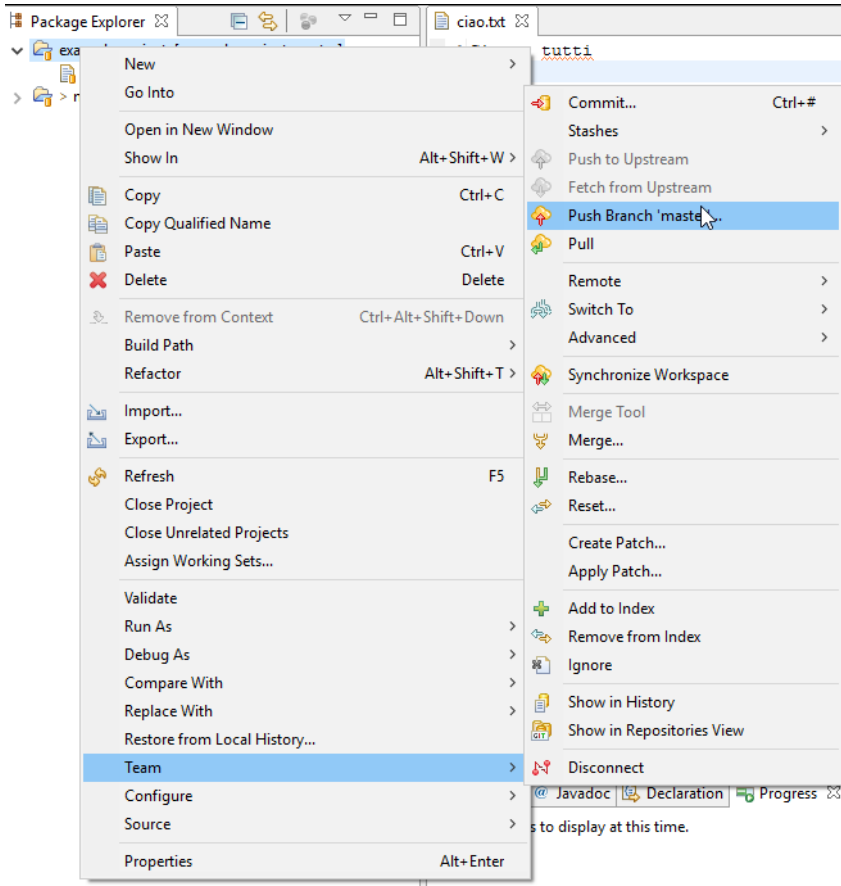
Add .gitignore: **None** | Add a license: **None** ⓘ

Create repository

New repository in Eclipse



Add remote & push in Eclipse

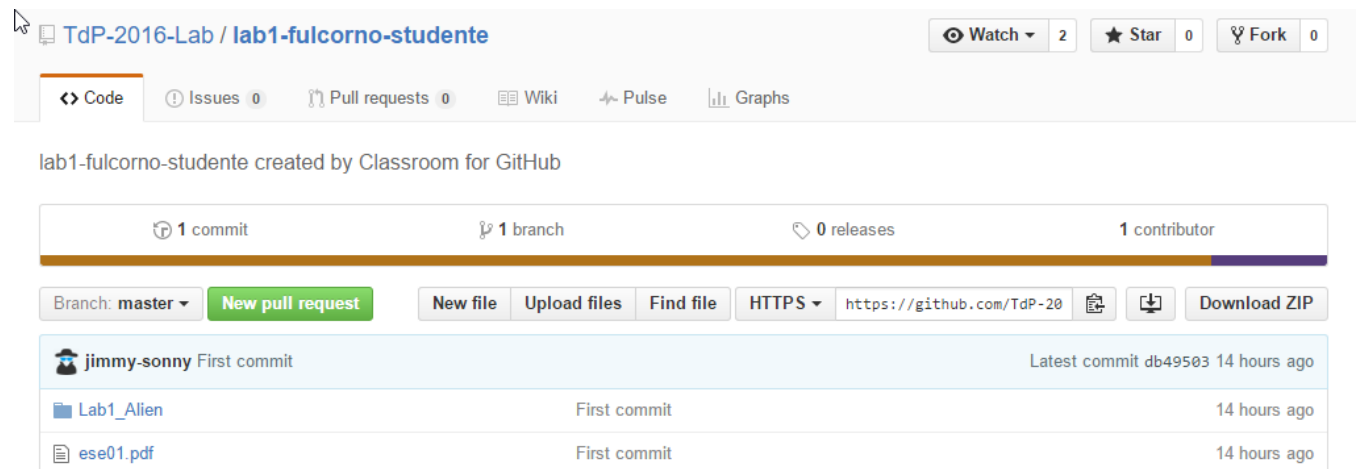


Workflow 2: “Work on a project”

1. “**Fork**” the project in GitHub (you make a copy in your repository)
2. **Clone** your project in Eclipse
3. Work on the project
4. **Commit** and **Push** the changes

Forking

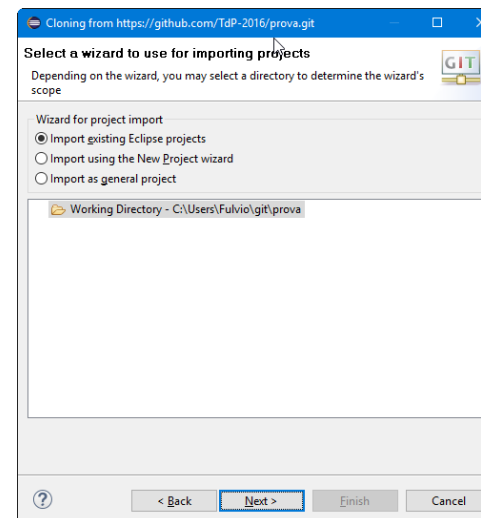
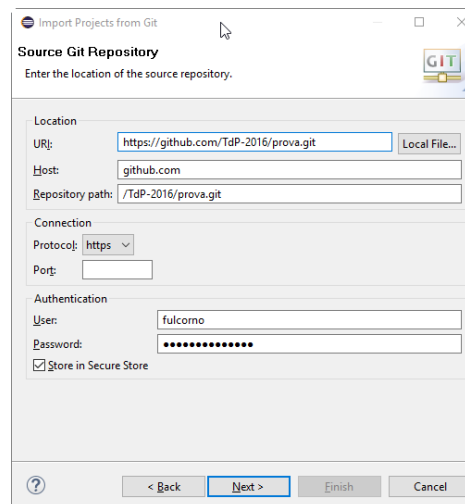
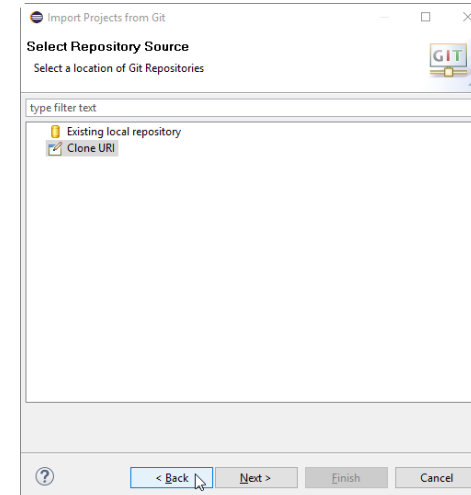
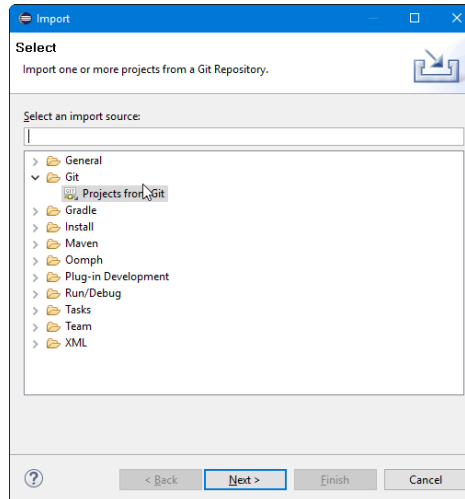
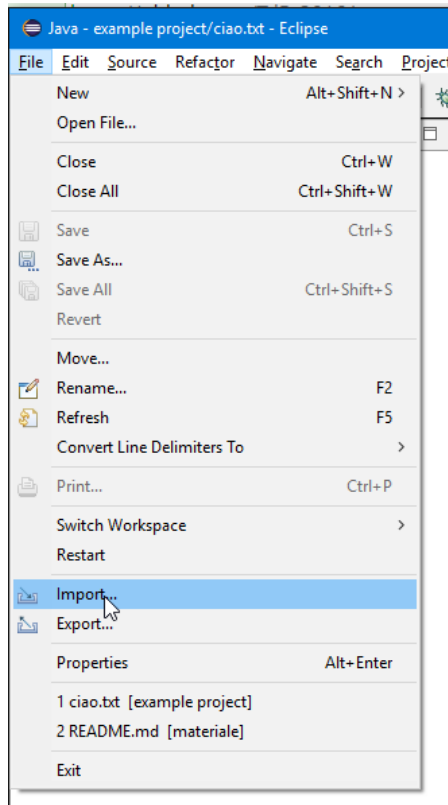
- “**Fork**” makes a private copy of some else’s repository
 - For example, the Lab projects
- You may clone, work, and commit on this repository



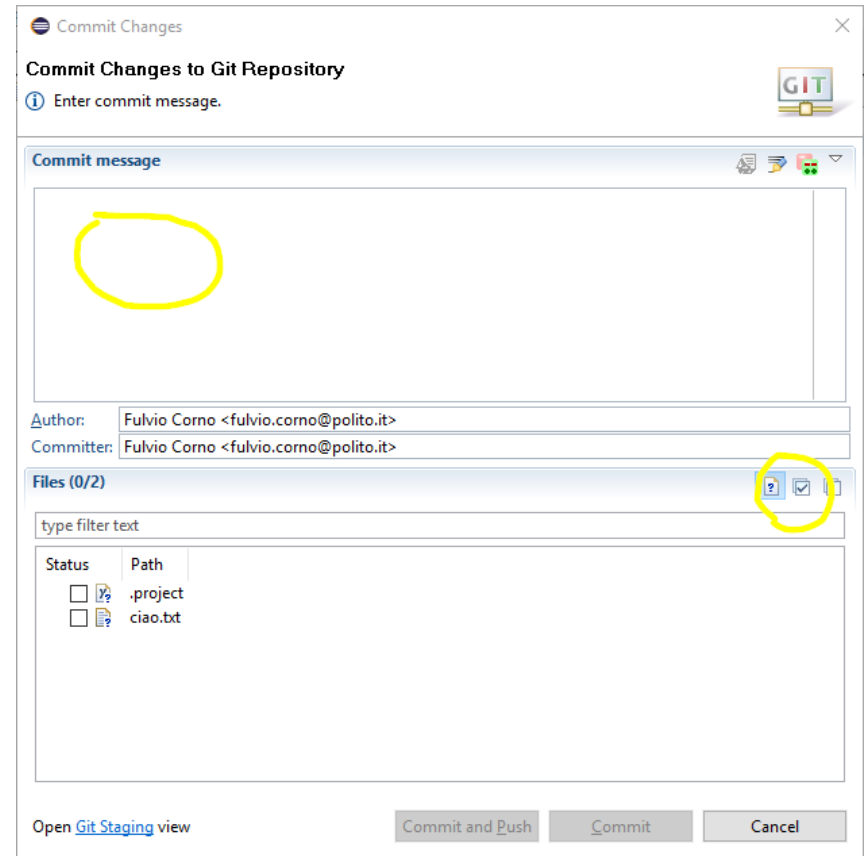
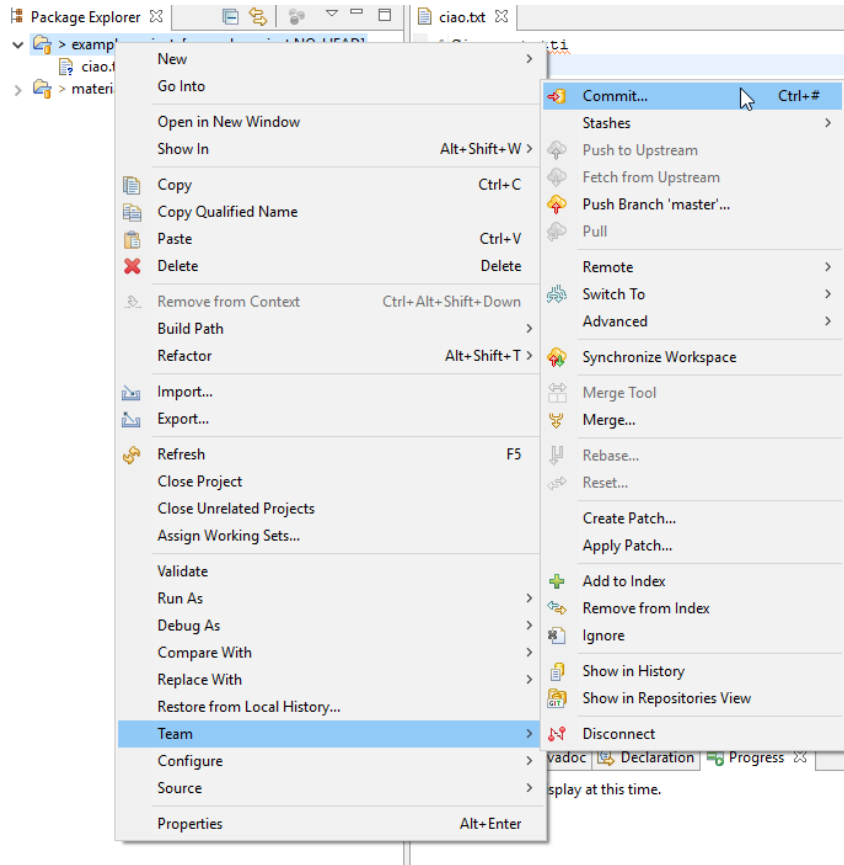
The screenshot shows the GitHub interface for the repository 'TdP-2016-Lab / lab1-fulcorno-studente'. At the top right, there are buttons for 'Watch' (2), 'Star' (0), and 'Fork' (0). Below these are navigation tabs for 'Code', 'Issues' (0), 'Pull requests' (0), 'Wiki', 'Pulse', and 'Graphs'. The repository name is followed by the text 'lab1-fulcorno-studente created by Classroom for GitHub'. A summary bar shows '1 commit', '1 branch', '0 releases', and '1 contributor'. Below this is a toolbar with 'Branch: master', a green 'New pull request' button, and buttons for 'New file', 'Upload files', 'Find file', 'HTTPS' (with a dropdown arrow), the URL 'https://github.com/TdP-20', a copy icon, a download icon, and a 'Download ZIP' button. The commit history table shows three entries:

Commit	Author	Message	Time
	jimmy-sonny	First commit	Latest commit db49503 14 hours ago
Lab1_Alien		First commit	14 hours ago
ese01.pdf		First commit	14 hours ago

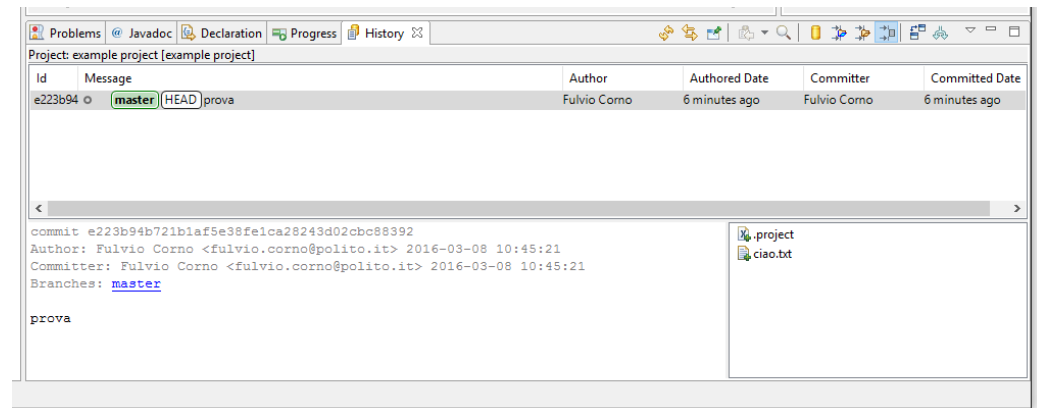
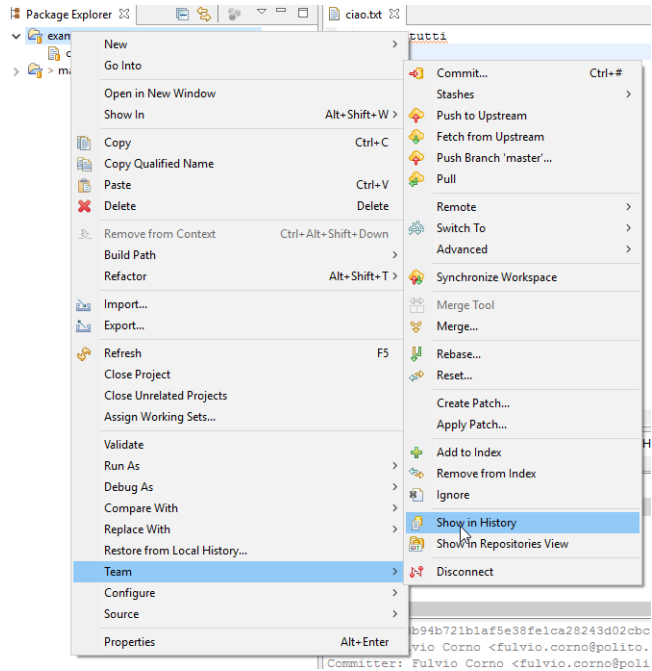
Cloning in Eclipse



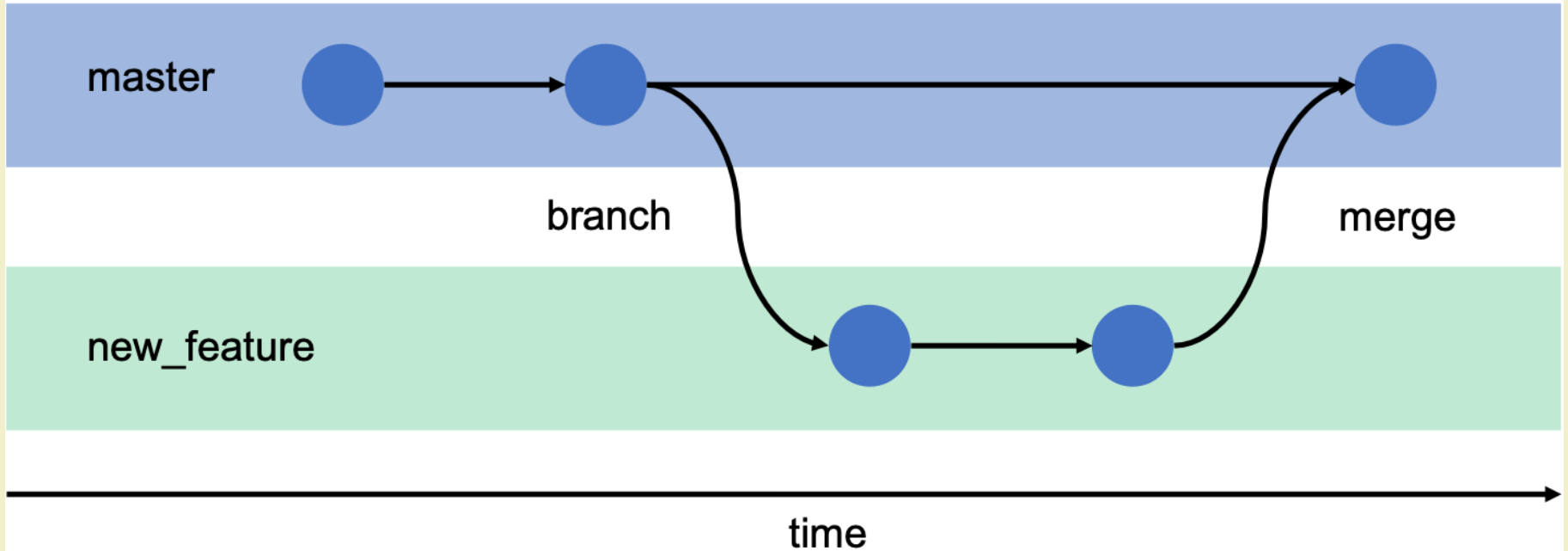
Commit in Eclipse



History in Eclipse



Branches



Branches... in brief

- used to develop features isolated from each other
- the *master* (or *main*) branch is the “default” branch when you create a repository
 - you should use other branches for development and merge them back to the master branch upon completion
- Branches can be local (your local repo) or may be pushed to GitHub

References

- Git Reference
 - <http://gitref.org/>
- Git - the simple guide
 - <http://rogerdudler.github.io/git-guide/>
- Git Documentation
 - <http://git-scm.com/docs>
- Pro Git (online book)
 - <http://git-scm.com/book>
- Version Control by Example (online book)
 - <http://www.ericssink.com/vcbe/>




References

- Try Git!
 - <http://try.github.io/>
- Various Git resources
 - <https://help.github.com/articles/what-are-other-good-resources-for-learning-git-and-github>
- A successful Git branching model
 - <http://nvie.com/posts/a-successful-git-branching-model/>
- Some Git (graphical) clients
 - <http://git-scm.com/downloads/guis>

References for Personal Access Tokens

- <https://github.blog/2020-12-15-token-authentication-requirements-for-git-operations/>
- <https://docs.github.com/en/github/authenticating-to-github/keeping-your-account-and-data-secure/creating-a-personal-access-token>
- [https://wiki.eclipse.org/EGit/User_Guide#GitHub Tutorial](https://wiki.eclipse.org/EGit/User_Guide#GitHub_Tutorial)

License

- This work is licensed under the Creative Commons “Attribution-NonCommercial-ShareAlike Unported (CC BY-NC-SA 3,0)” License.
- You are free:
 - to **Share** - to copy, distribute and transmit the work
 - to **Remix** - to adapt the work
- Under the following conditions:
 - **Attribution** - You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work). 
 - **Noncommercial** - You may not use this work for commercial purposes. 
 - **Share Alike** - If you alter, transform, or build upon this work, you may distribute the resulting work only under the same or similar license to this one. 
- To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/3.0/>