AESYS ACADEMY

08-04-2022

ACADEMY
JAVA

Agenda

AESYS ACADEMY

Java

ACADEMY
JAVA

Agenda

- Introduzione a GIT ed ai sistemi di versionamento
- Getting Started
- Le basi di GIT
- Repository Remoti
- .gitignore

Introduzione a GIT ed ai sistemi di versionamento

AESYS ACADEMY

Java

ACADEMY
JAVA

Cos’è un VCS (Version Control System)

Un VCS mantiene una base di dati che conserva le modifiche effettuate sui file tracciati

- **locale**: i file tracciati risiedono sulla propria macchina
- **centralizzato**: i file tracciati risiedono solamente su un server centrale
- **decentralizzato (o distribuito)**: la base di dati e i file tracciati risiedono sulle macchine di ciascun utente

Cos’è GIT

GIT è un VCS *decentralizzato*, che tiene traccia delle modifiche ai file tramite snapshot

- la maggior parte delle operazioni sono svolte in locale sulle macchine degli utilizzatori;
- ogni modifica viene salvata con un controllo di integrità;

Getting Started

ACADEMY
JAVA

Installare la CLI

Per utilizzare GIT all'interno del proprio sistema operativo è necessario installare la sua CLI disponibile sul sito ufficiale per tutte le distribuzioni (Windows, Linux, MacOS).

Una volta installata la CLI non dobbiamo fare nient'altro, siamo pronti per operare utilizzando **git** direttamente dal nostro terminale.

Le basi di GIT

ACADEMY
JAVA

Inizializzare un’area di progetto

Una directory i cui file siano tracciati da GIT è chiamata repository;
Per crearne una locale basta solo:

- spostarsi nella directory radice del progetto che si vuole tracciare
- eseguire il seguente comando direttamente dal nostro terminale: **$ git init**

Inizializzare un’area di progetto

Configurare GIT:

- **$ git config user.name** (Imposta il nome utente che sarà associato alle modifiche)
- **$ git config user.email** (Imposta anche una mail da associare a un nome utente)
- **$ git config core.editor** (Scegli l’editor predefinito da usare con GIT)

Clonare un progetto online

Per clonare un progetto che risiete su una macchina remota possiamo utilizzare il comando:

- $ **git clone {url}**

In questo modo verrà copiato sulla propria macchina anche lo storico delle modifiche di tutti i file del repository remoto.

Modifiche e Staging Area

Ho capito come creare il mio repository locale o come scaricarlo, ma come faccio a tenere traccia delle modifiche?
GIT gestisce i file permettendo all'utente di assegnargli uno stato, nei modi seguenti:

- **staged**: ho chiesto a GIT di «mettere da parte» lo snapshot di un file col comando **$ git add**
- **modified**: ho chiesto a GIT di tracciare un file con **$ git add** , dopodiché ho modificato quel file
- **committed**: ho chiesto a GIT di prendere il contenuto della staging area e di spostarlo nel database degli snapshot, tramite il comando **$ git commit**

Cos’è un Branch

I rami (branch o più estesamente “*ramificazioni*”) vengono utilizzati in git per l’implementazione di funzionalità tra loro isolate, cioè sviluppate in modo indipendente l’una dall’altra ma a partire dalla medesima radice.

Commit

Per salvare permanentemente gli snapshot dei file contenuti nella staging area, si usa il comando **$ git commit**

- con il comando –m «» viene aggiunto un messaggio per identificare la commit
- gli snapshot nella staging area saranno spostati nella cartella .git, e i file associati saranno marcati come «unmodified» Possiamo controllare in ogni momento lo stato dei file tracciati e nella staging area
- **$ git status:** mostra lo stato dei file tracciati, indicando le modifiche e il contenuto della staging area (così sapremo cosa finirà nel commit)
- **$ git diff <--staged>:** mostra in modo specifico le differenze dei contenuti dei file «modified» ma non «staged» rispetto all'ultimo commit

Push & Pull

Dopo aver associato dei repository remoti a un progetto, possiamo caricare o scaricare modifiche da e verso di essi:

- col comando **$ git pull** scarichiamo dal repository remoto i commit che non abbiamo in locale
- col comando **$ git push** carichiamo le modifiche al progetto sul repository remoto

Conflitti e Merging

Nel caso in cui due commit contengano modifiche allo stesso file, bisogna scegliere manualmente la modifica da tenere prima di poter caricare o scaricare entrambi i commit nello stesso repository

Comandi

command	description
`git clone` ***`url [dir]`***	copy a Git repository so you can add to it
`git add` ***`file`***	adds file contents to the staging area
`git commit`	records a snapshot of the staging area
`git status`	view the status of your files in the working directory and staging area
`git diff`	shows diff of what is staged and what is modified but unstaged
`git help` ***`[command]`***	get help info about a particular command
`git pull`	fetch from a remote repo and try to merge into the current branch
`git push`	push your new branches and data to a remote repository
others: `init, reset, branch, checkout, merge, log, tag`	

.gitignore

AESYS ACADEMY

ACADEMY
JAVA

Cos’è il file .gitignore

. **gitignore** dice a **git** quali file (o modelli) dovrebbe ignorare. Di solito viene utilizzato per evitare il commit di file temporanei dalla directory di lavoro che non sono utili ad altri collaboratori, come prodotti di compilazione, creazione di file IDE temporanei, ecc.

AESYS ACADEMY

GRAZIE

ACADEMY
JAVA