**TÌM HIỂU GITHUB**

**I.Thế nào là Version Control System (VCS):**

Là một hệ thống lưu giữ các phiên bản của [mã nguồn](https://vi.wikipedia.org/wiki/M%C3%A3_ngu%E1%BB%93n) của sản phẩm phần mềm, giúp các [lập trình viên](https://vi.wikipedia.org/wiki/L%E1%BA%ADp_tr%C3%ACnh_vi%C3%AAn) có thể dễ dàng lấy lại phiên bản mong muốn.

Hệ thống này có thể được sử dụng bởi một nhóm các lập trình viên, mỗi thành viên trong nhóm thường không được phép thay đổi mã nguồn của các thành viên khác, mà chỉ có thể xem. VCS cho phép người quản trị phân chia các [tập tin](https://vi.wikipedia.org/wiki/T%E1%BA%ADp_tin) cho từng thành viên tương ứng. Nó cũng cho phép các thành viên chia sẻ một số tập tin cho nhau trong khi phát triển. Các thành viên có thể phát hiện lỗi và sửa lỗi thuận tiện trong VCS. Trưởng nhóm phải có nhiệm vụ cập nhật lại nội dung của các tập tin đó. VCS giúp cho công việc này được thực hiện một cách tự động.

Khi các thành viên hiệu chỉnh mã của cùng một tập tin tại cùng một thời điểm, để tránh sửa đổi mâu thuẫn, họ sẽ phải so sánh xem có gì khác biệt giữa các sửa đổi của các thành viên hay không. VCS giúp cho việc này được thực hiện tự động.

Có 3 loại VCS:

+Local Version Control System(cũ nhất)

+Cetralized Version Cotrol System

+Distributed Version Control System(hiện đại nhất GIT nằm ở loại này)  
  
**II.Tổng quan về GIT:**

Bạn là lập trình viên, và đôi khi bạn muốn đưa về trạng thái trước khi quậy phá của file code nào đó? Cách đơn giản nhất đó là sao chép lại file trước khi chỉnh sửa. Trường hợp dùng phương pháp này thì sẽ phải thường xuyên thực hiện việc thêm ngày đã thay đổi vào tên thư mục hay file. Tuy nhiên, việc tự mình sao chép file mỗi lần chỉnh sửa thì sẽ rất vất vả, và cũng dễ xảy ra nhầm lẫn.  
Và để giải quyết những vấn đề này thì các hệ thống quản lý phiên bản như Git đã được ra đời.

*Sơ lược lịch sử về GIT:*

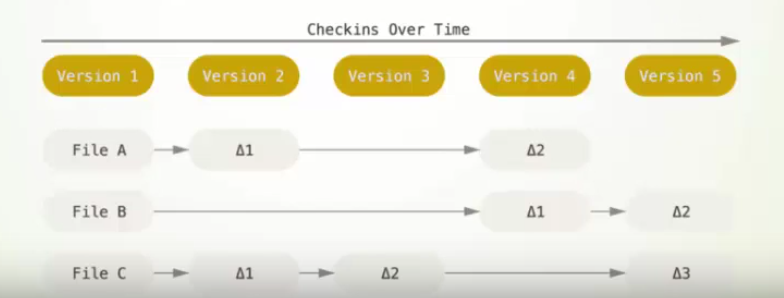
2002, cha để của hệ điều hành Linux bắt đầu một dự án là BitKeeper với một công ty DVCS

2005, xảy ra mâu thuẫn dự án BitKeeper đã bị lấy lại, cộng đồng Linux đã xây dựng một tool quản lý source code dựa trên những kinh nghiệm làm việc trên Bitkeeper trước đó.

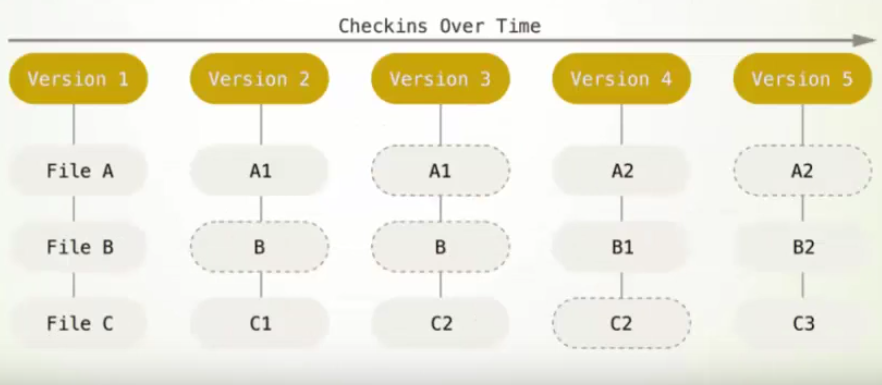
*GIT căn bản:*

Git là một trong những Hệ thống Quản lý Phiên bản Phân tán, vốn được phát triển nhằm quản lý mã nguồn (source code) của Linux.  
  
Trên Git, ta có thể lưu trạng thái của file dưới dạng lịch sử cập nhật. Vì thế, có thể đưa file đã chỉnh sửa một lần về trạng thái cũ hay có thể biết được file đã được chỉnh sửa chỗ nào.

Sự tư duy của GIT khác biệt so với các hệ thống VCS khác (Subversion, CVS, Perfore…)



**Cách lưu file của các VCS khác**

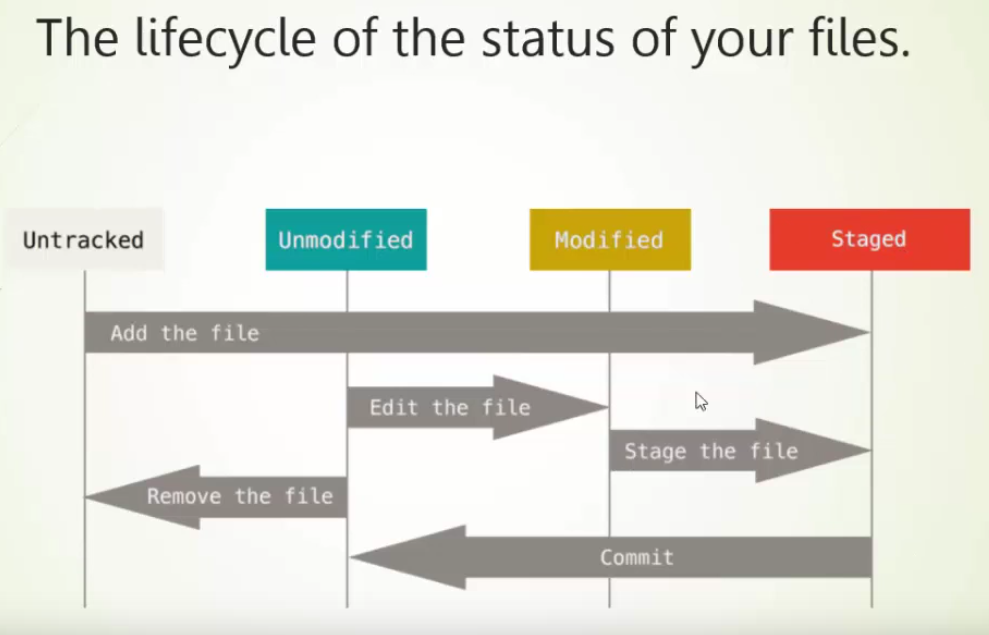
****

**Cách lưu file của GIT**

Sự khác nhau cơ bản giữa Git với bất kỳ VCS nào khác (bao gồm Subversion và tương tự là cách Git "nghĩ" về dữ liệu. Về mặt lý thuyết mà nói, phần lớn hệ thống khác lưu trữ thông tin dưới dạng danh sách các tập tin được thay đổi. Các hệ thống này (CVS, Subversion, Perforce, Bazaar,...) coi thông tin được lưu trữ như là một tập hợp các tập tin và các thay đổi được thực hiện trên mỗi tập tin theo thời gian, được minh hoạ trong hình 1-4.

Hình 1-4. Các hệ thống khác hướng tới lưu trữ tập tin dưới dạng các thay đổi so với bản cơ sở của mỗi tập tin.

Git không nghĩ hoặc xử lý dữ liệu theo cách này. Mà thay vào đó Git coi dữ liệu của nó giống như một tập hợp các "ảnh" (snapshot) của một hệ thống tập tin nhỏ. Mỗi lần bạn "commit", hoặc lưu lại trạng thái hiện tại của dự án trong Git, về cơ bản Git "chụp một bức ảnh" ghi lại nội dung của tất cả các tập tin tại thời điểm đó và tạo ra một tham chiếu tới "ảnh" đó. Để hiệu quả hơn, nếu như tập tin không có sự thay đổi nào, Git không lưu trữ tập tin đó lại một lần nữa mà chỉ tạo một liên kết tới tập tin gốc đã tồn tại trước đó. Git thao tác với dữ liệu giống như Hình 1-5.

****

Kho Repo (Repository)

Repository hay được gọi tắt là Repo, đơn giản là nơi chứa tất cả những thông tin cần thiết để duy trì và quản lý các sửa đổi và lịch sử của toàn bộ project. Trong Repo có 2 cấu trúc dữ liệu chính là Object Store và Index. Tất cả dữ liệu của Repo đèu được chứa trong thư mục bạn đang làm việc dưới dạng folder ẩn có tên là .git

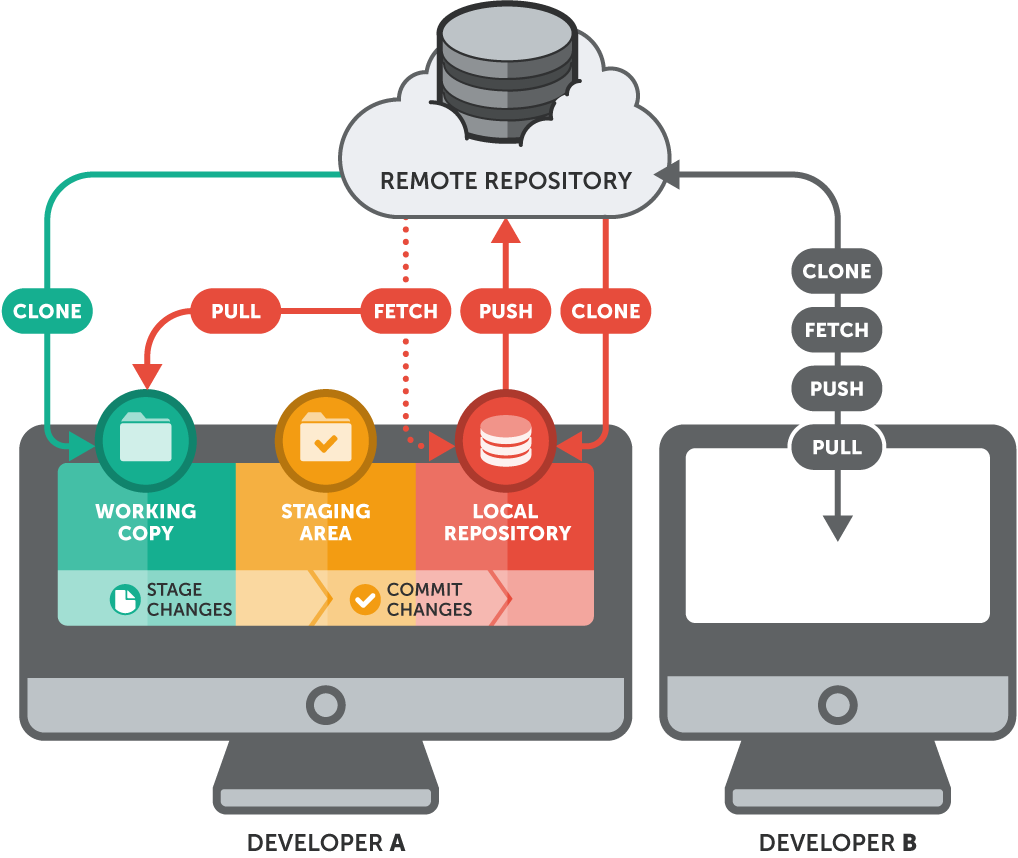
## Remote repository và local repository

Đầu tiên, repository của Git được phân thành 2 loại là remote repository và local repository.

**Remote repository:** Là repository để chia sẻ giữa nhiều người và bố trí trên server chuyên dụng.

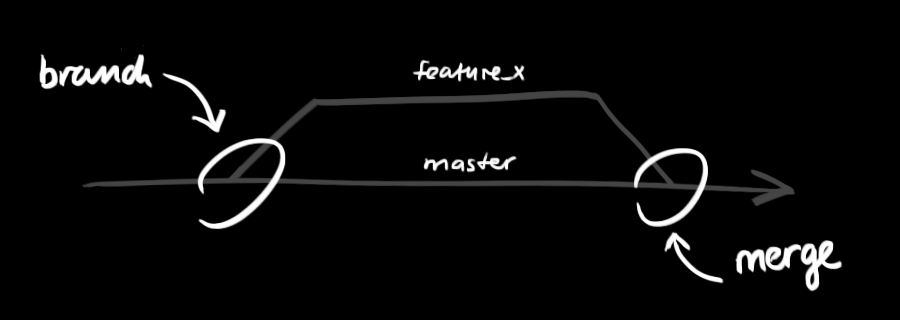
**Local repository:** Là repository bố trí trên máy của bản thân mình, dành cho một người dùng sử dụng.

Do repository phân thành 2 loại là local và remote nên với những công việc bình thường thì có thể sử dụng local repository. Khi muốn public nội dung công việc mà mình đã làm trên local repository, thì ta sẽ upload lên remote repository rồi public. Thêm nữa, thông qua remote repository bạn cũng có thể lấy về nội dung thay đổi của người khác.



Nhánh (Branch)

Đây là một trong những thế mạnh của git là nhánh. Với git, việc quản lý nhánh rất dễ dàng. Mỗi nhánh trong Git gần giống như một workspace. Việc nhảy vào một nhánh để làm việc trong đó tương tự việc chuyển qua ngữ cảnh làm việc mới, và sau đó có thể nhanh chóng quay lại ngữ cảnh cũ.  
  
Nhánh (branch) được dùng để phát triển tính năng mới mà không làm ảnh hưởng đến code hiện tại.  
  
Nhánh master là nhánh “mặc định” khi bạn tạo một repository.  
Nhánh master thông thường là nhánh chính của ứng dụng. Ví dụ bạn thử nghiệm một tính năng mới và muốn không ảnh hưởng đến code chính bạn có thể tạo một nhánh mới và sau khi xong sẽ hợp nhất lại với nhánh master. Việc hợp nhất 2 nhánh lại được gọi là merge.

****

## Trộn (Merge)

Trộn source từ một nhánh khác vào nhánh hiện tại.  
Chú ý:

Kiểm tra branch hiện đang làm việc trước khi merge

Phải đẩy tất cả những thay đổi dưới máy local lên Git trước khi merge

Trước khi merge phải lấy hết những thay đổi mới nhất của các branch khác, hay ít nhất là branch cần merge về máy

Merge thành công thì nên đẩy source lên lại lên server

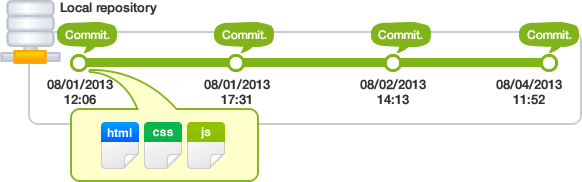
Nên merge bằng GUI tool.

Commit

Để ghi lại việc thêm/thay đổi file hay thư mục vào repository thì sẽ thực hiện thao tác gọi là Commit.

Khi thực hiện commit, trong repository sẽ tạo ra commit (hoặc revision) đã ghi lại sự khác biệt từ trạng thái đã commit lần trước với trạng thái hiện tại.

Commit này đang được chứa tại repository, các commit nối tiếp với nhau theo thứ tự thời gian. Bằng việc lần theo commit này từ trạng thái mới nhất thì có thể biết được lịch sử thay đổi trong quá khứ hoặc nội dung thay đổi đó.

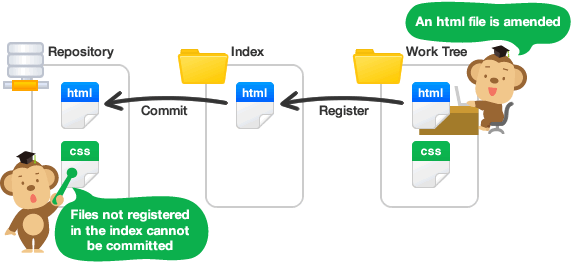


Mỗi commit đều có yêu cầu phải có commit message, để giải thích commit này là bạn đã làm gì trong này.

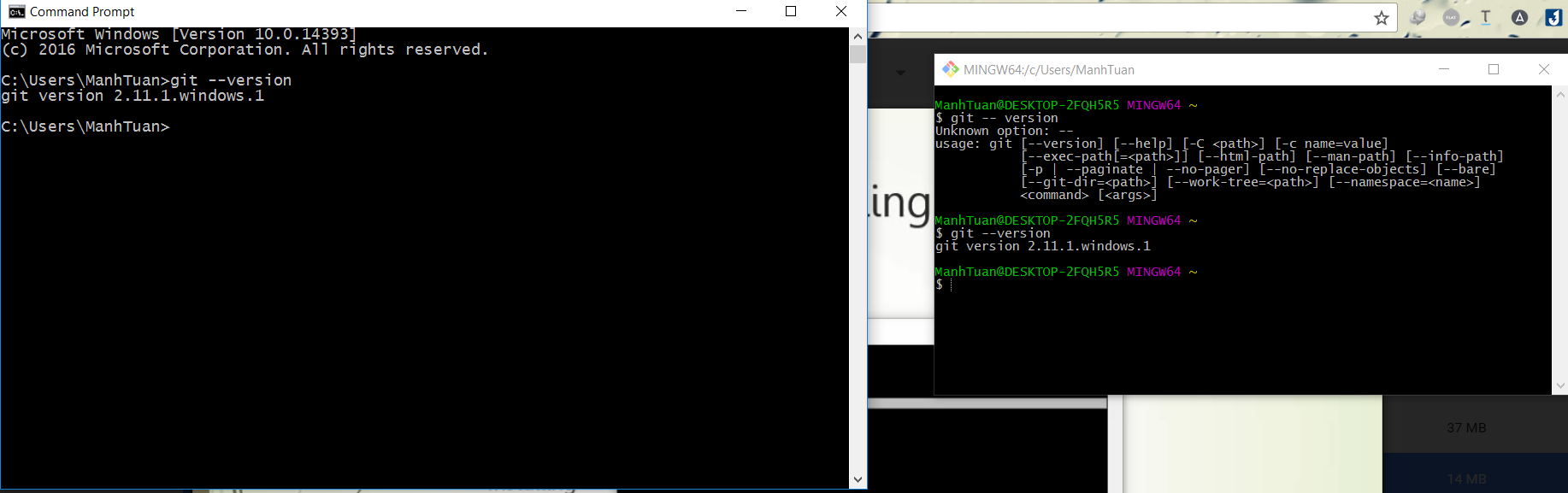
Working Tree và Index

Trên Git, những thư mục được đặt trong sự quản lý của Git mà mọi người đang thực hiện công việc trong thực tế được gọi là working tree.

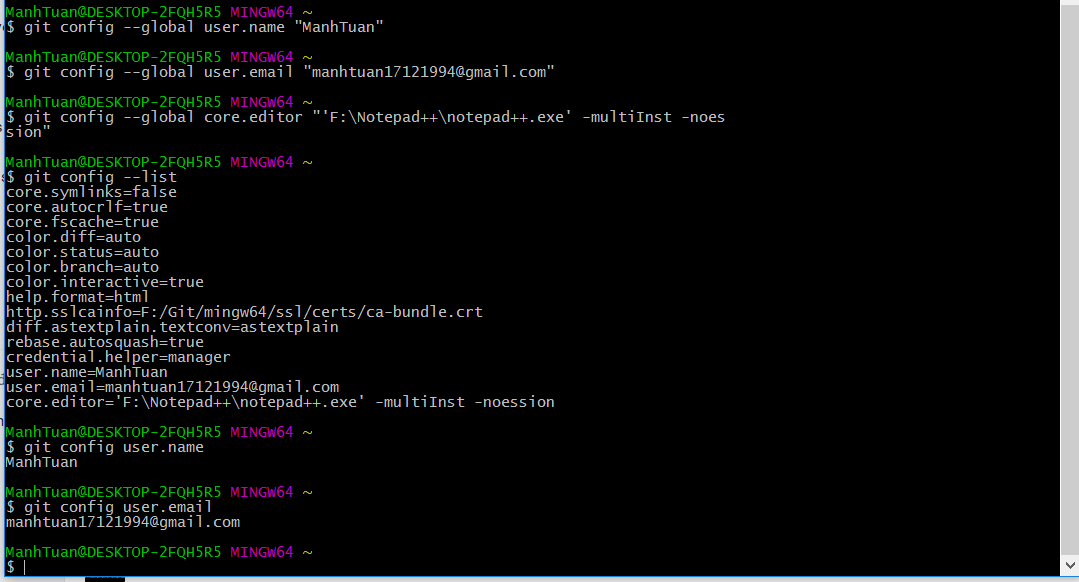
Và trên Git, giữa repository và working tree tồn tại một nơi gọi là index. Index là nơi để chuẩn bị cho việc commit lên repository.



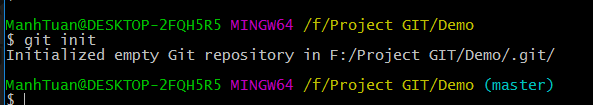
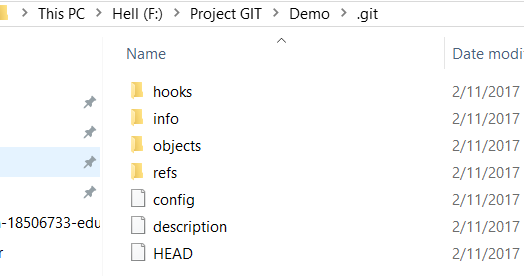
Trên Git, khi đã thực hiện commit thì trạng thái sẽ không được ghi trực tiếp trong repository từ working tree, mà sẽ ghi trạng thái đã được thiết lập của index được xây dựng ở giữa đó. Vì thế, để ghi lại trạng thái của file bằng commit thì trước hết cần thông báo file trong index.



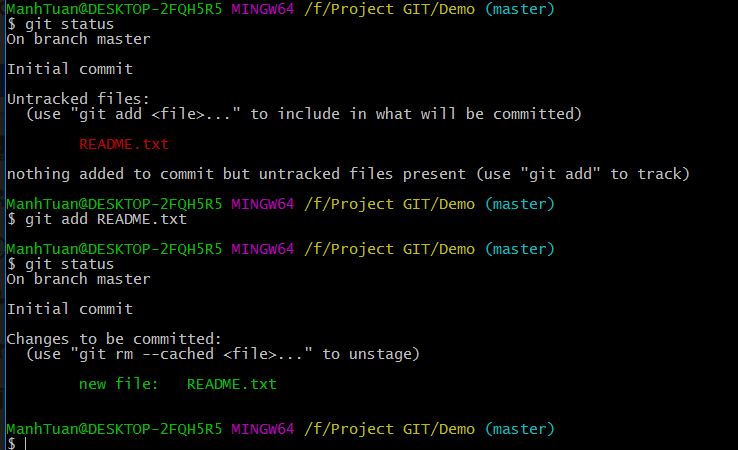
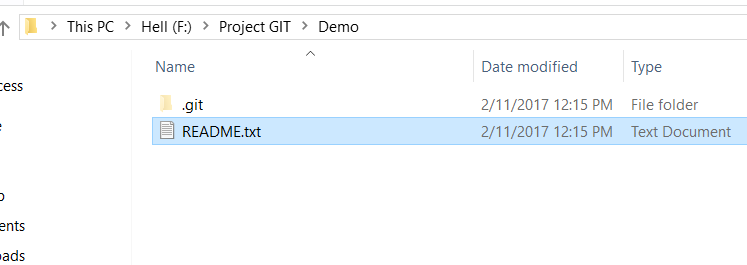
**Cài đặt xong GIT**



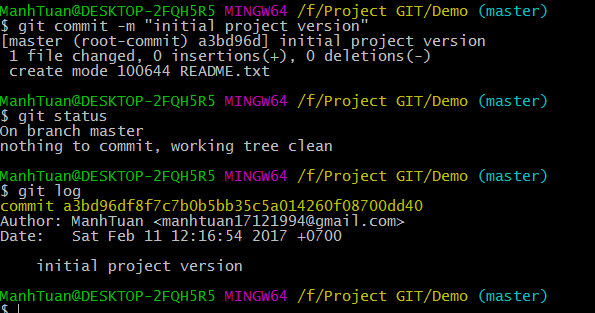
**Cấu hình GIT cho lần đầu sử dụng**

****

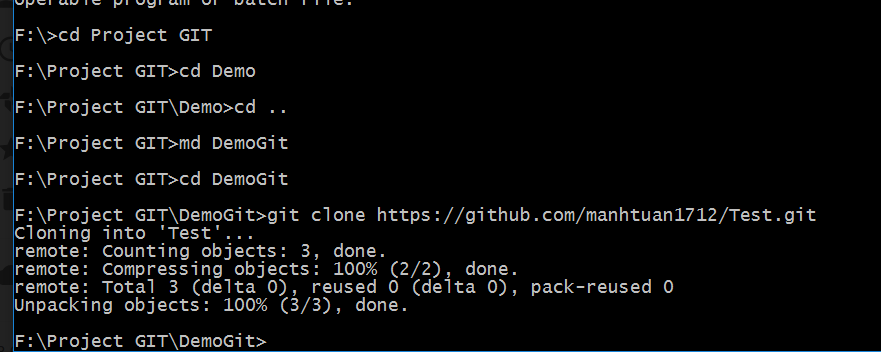
**Tạo một Repo**

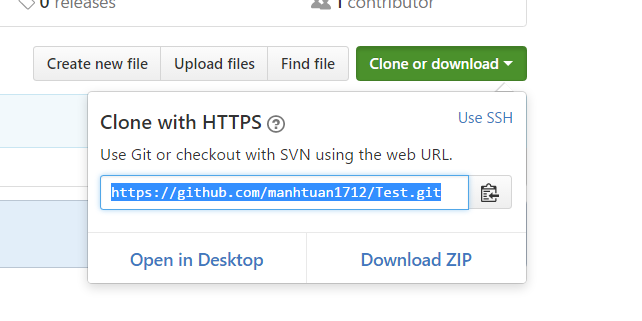
****

**Tạo và commit file README.txt**

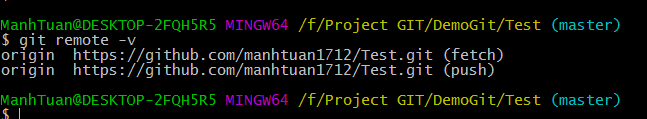
****

**Kiểm tra lịch sử commit**

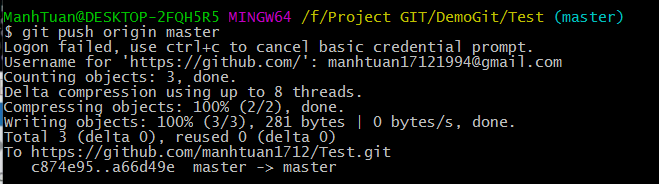
****

****

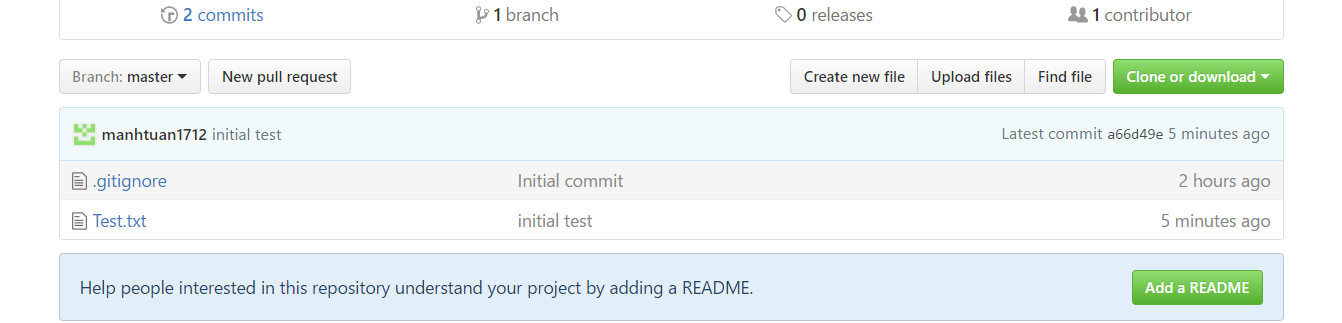
**Lấy một Repo bằng đưởng dẫn URL hoặc SSH address**

****

**Kiểm tra địa chỉ Remote Repo**

****

**Tạo một file Test.txt push lên Remote Repo**

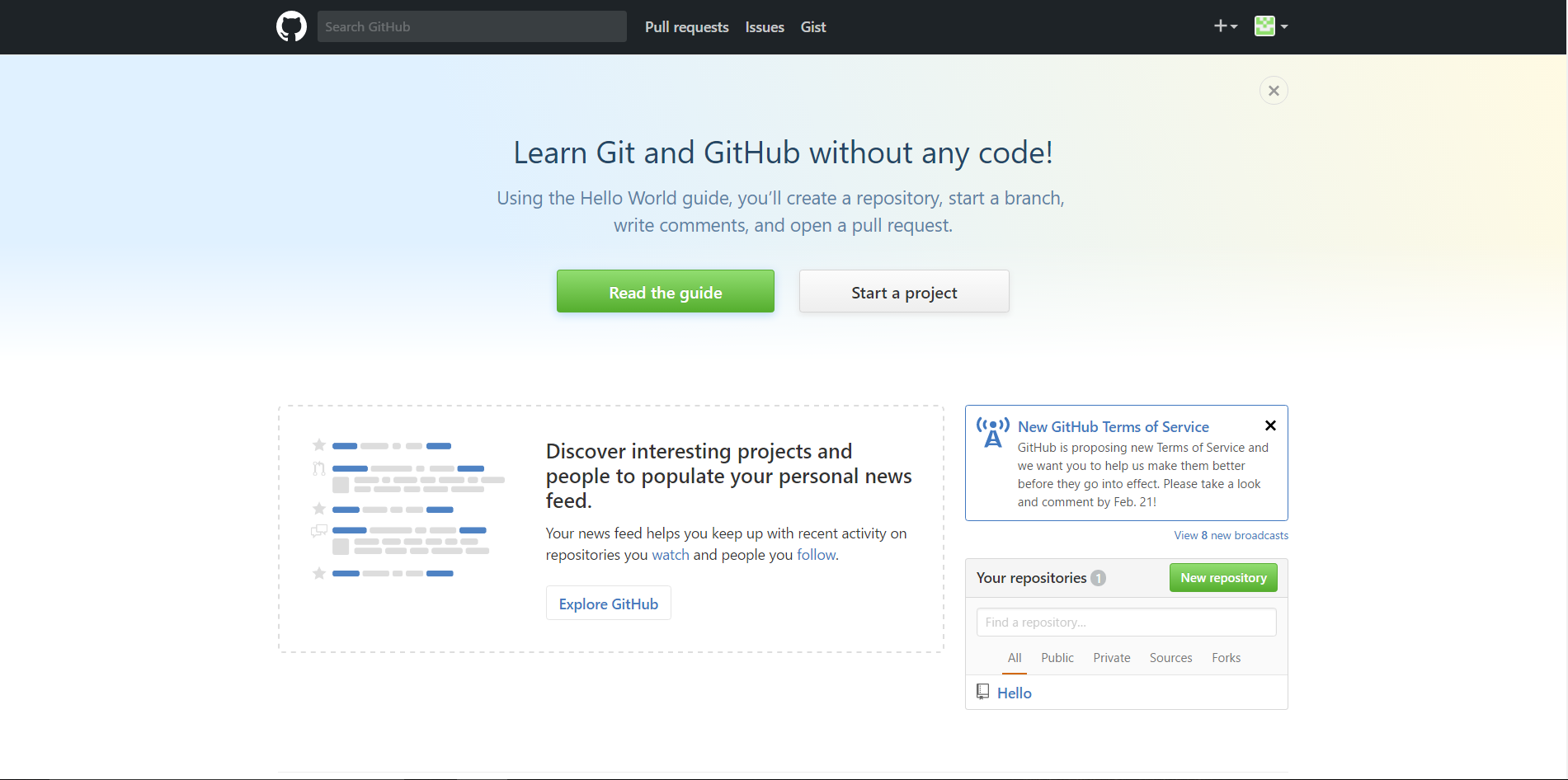
****

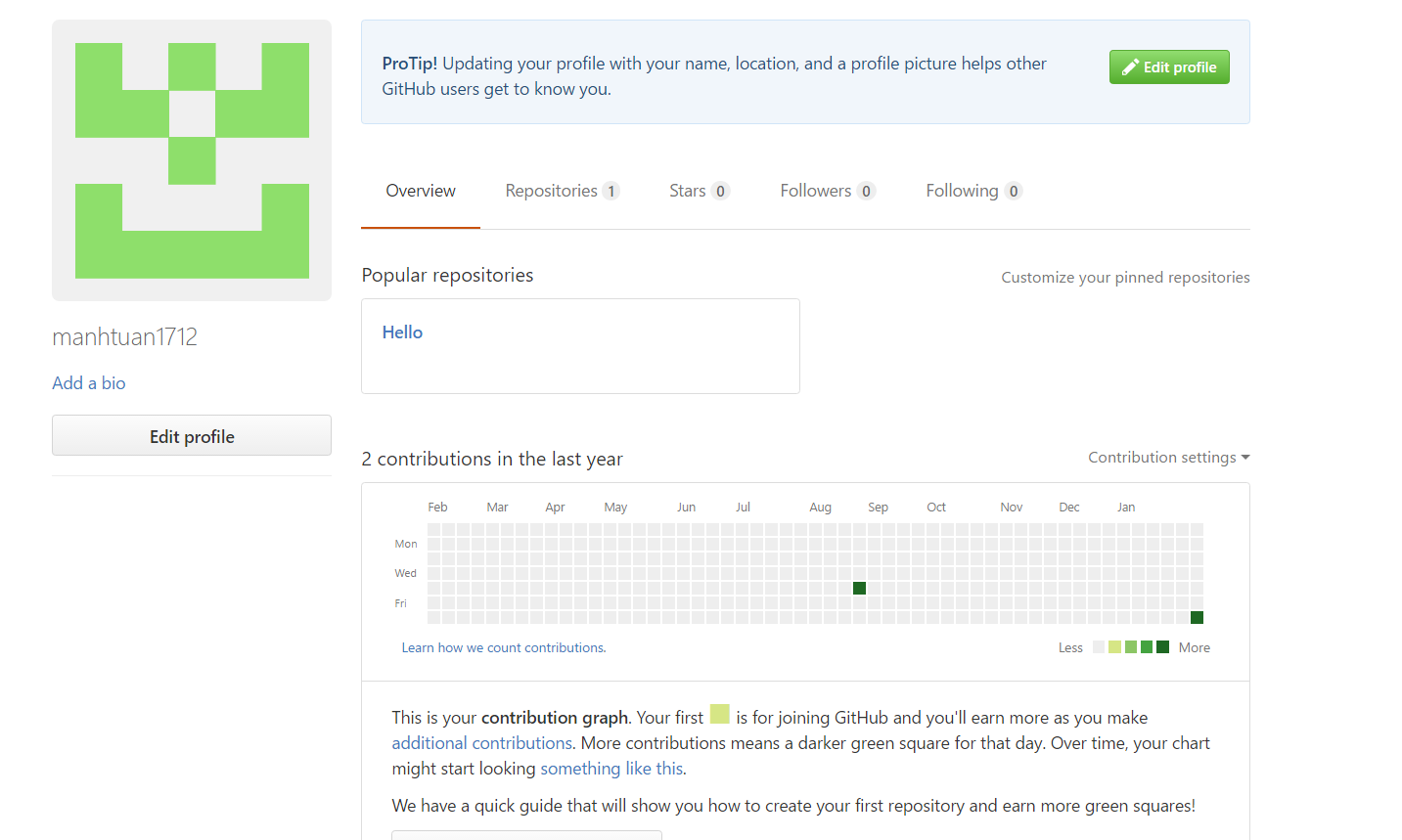
**Kết quả**

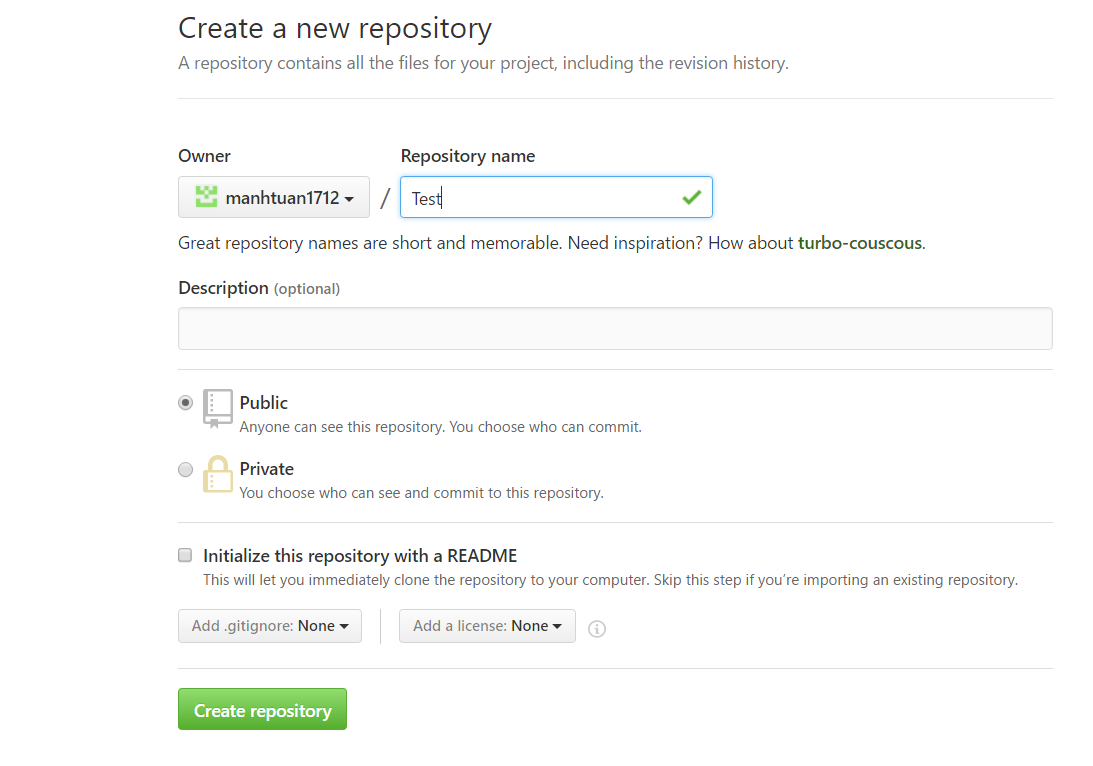
**III.GITHUB:**

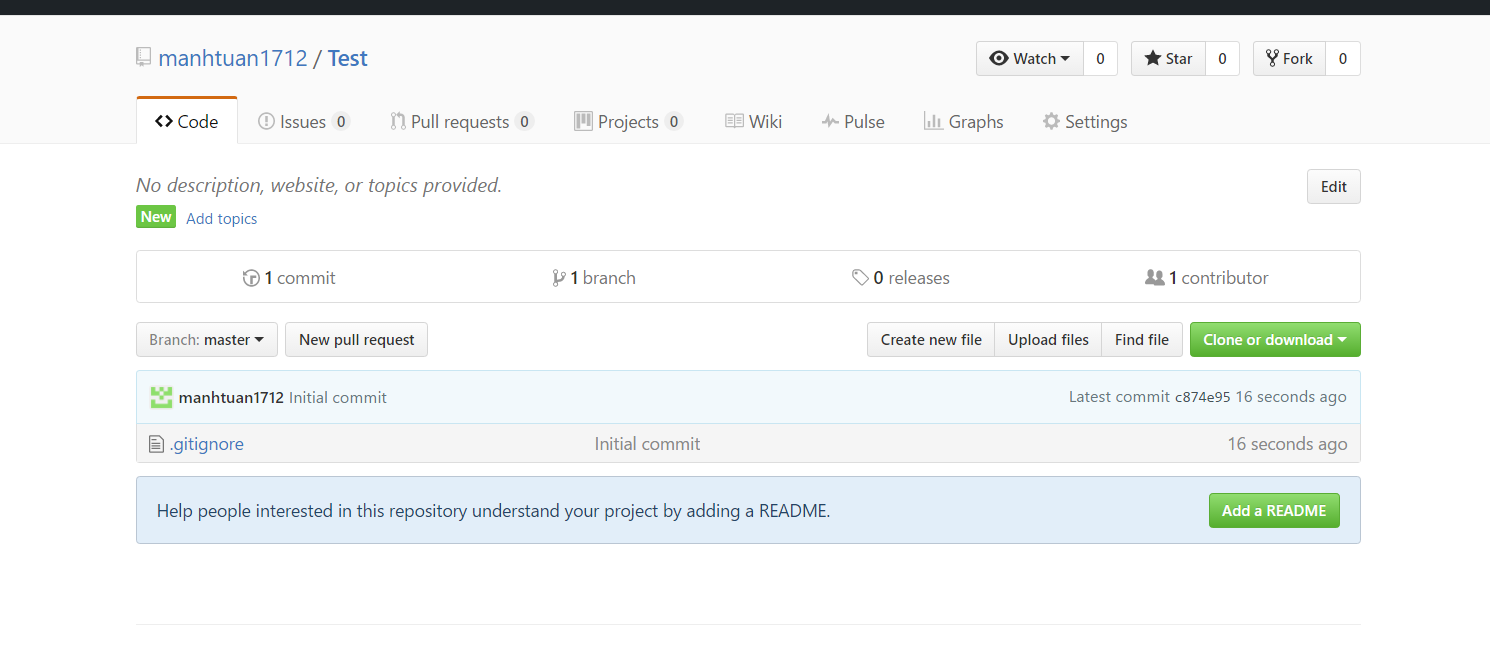
**GitHub** là một dịch vụ cung cấp [kho lưu trữ mã nguồn](https://vi.wikipedia.org/w/index.php?title=Kho_l%C6%B0u_tr%E1%BB%AF_m%C3%A3_ngu%E1%BB%93n&action=edit&redlink=1) [Git](https://vi.wikipedia.org/wiki/Git_(ph%E1%BA%A7n_m%E1%BB%81m)) dựa trên nền web cho các dự án phát triển phần mềm. GitHub cung cấp cả phiên bản trả tiền lẫn miễn phí cho các tài khoản. Các dự án [mã nguồn mở](https://vi.wikipedia.org/wiki/Ph%E1%BA%A7n_m%E1%BB%81m_ngu%E1%BB%93n_m%E1%BB%9F) sẽ được cung cấp kho lưu trữ miễn phí. Tính đến tháng 4 năm 2016, GitHub có hơn 14 triệu người sử dụng với hơn 35 triệu kho mã nguồn[[3]](https://vi.wikipedia.org/wiki/GitHub#cite_note-3), làm cho nó trở thành máy chủ chứa mã nguồn lớn trên thế giới

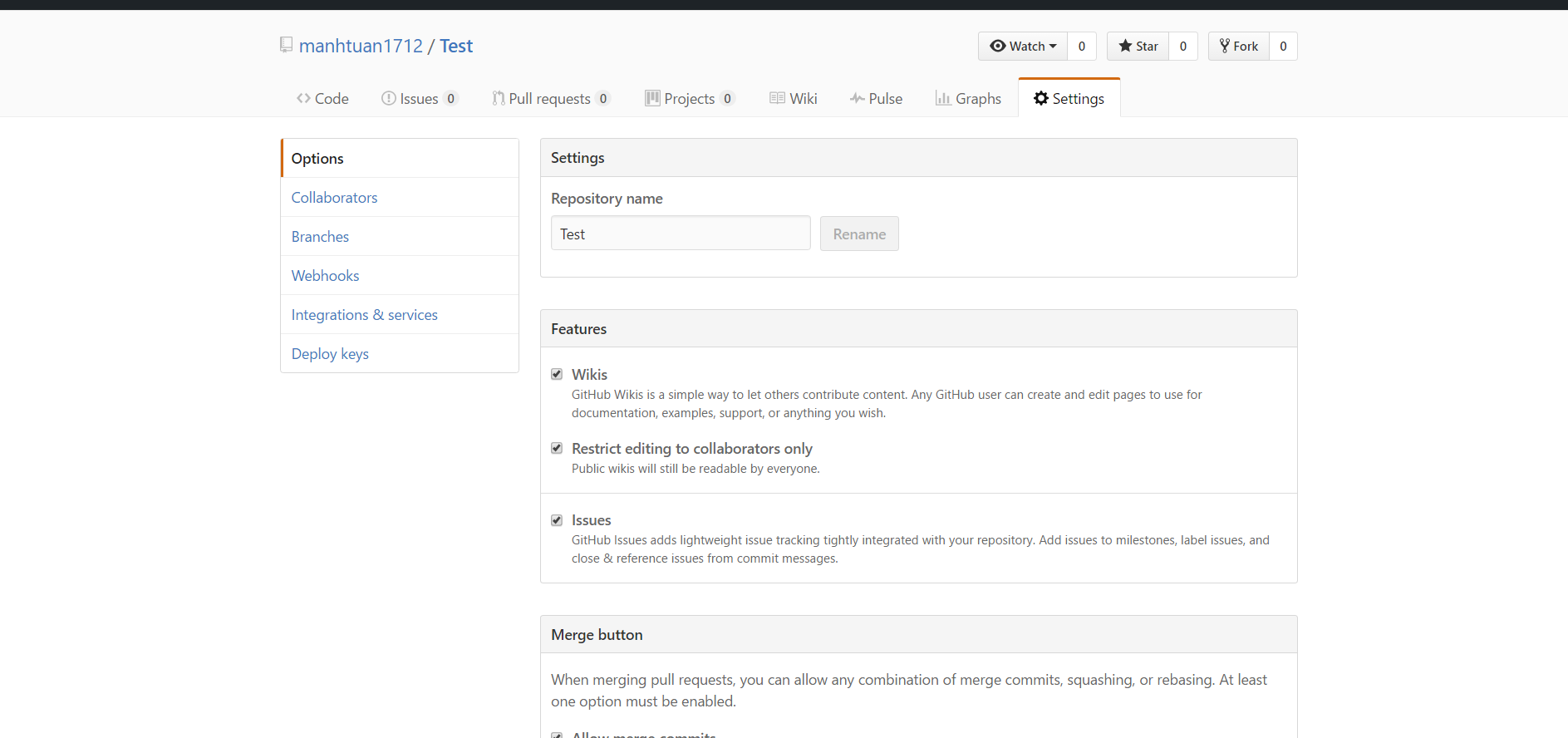
****

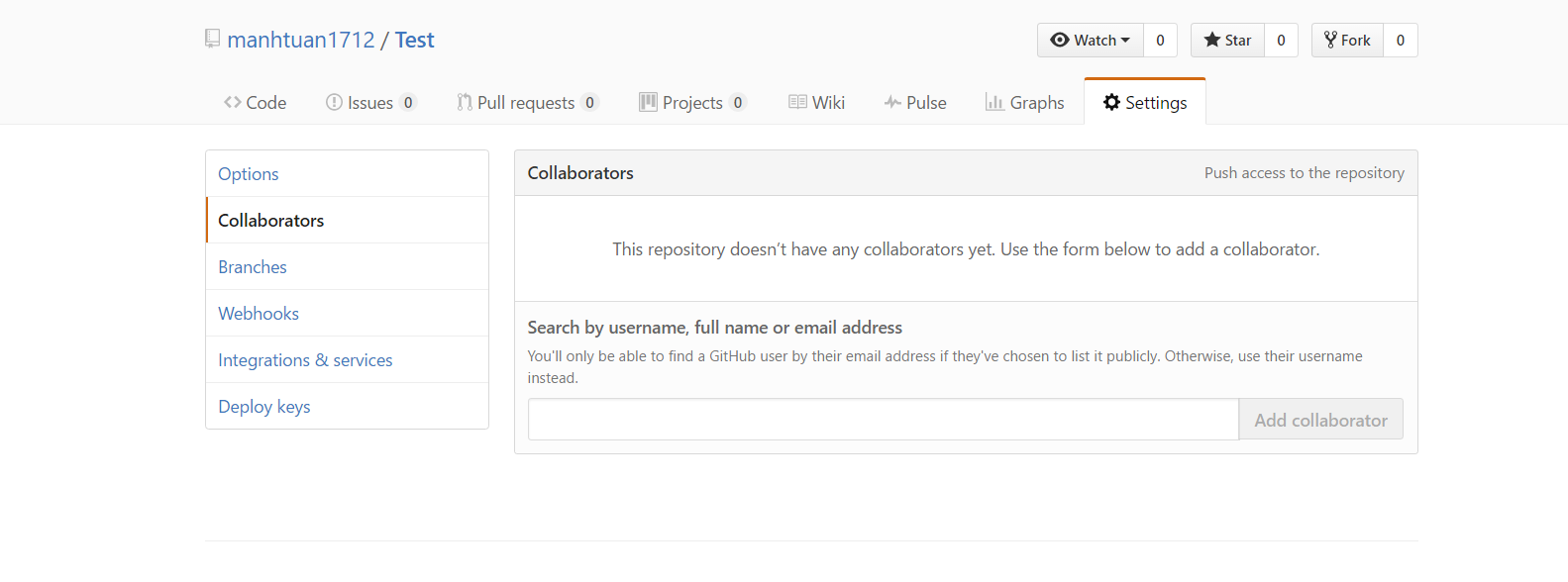
****

****

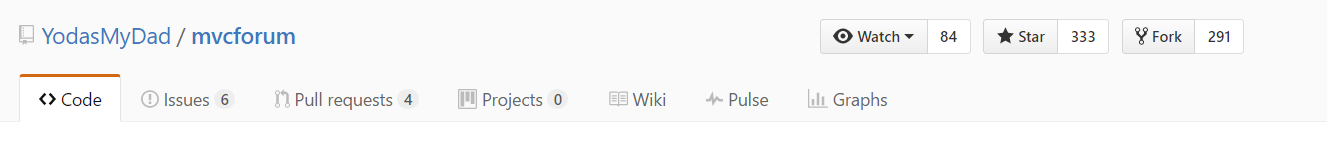
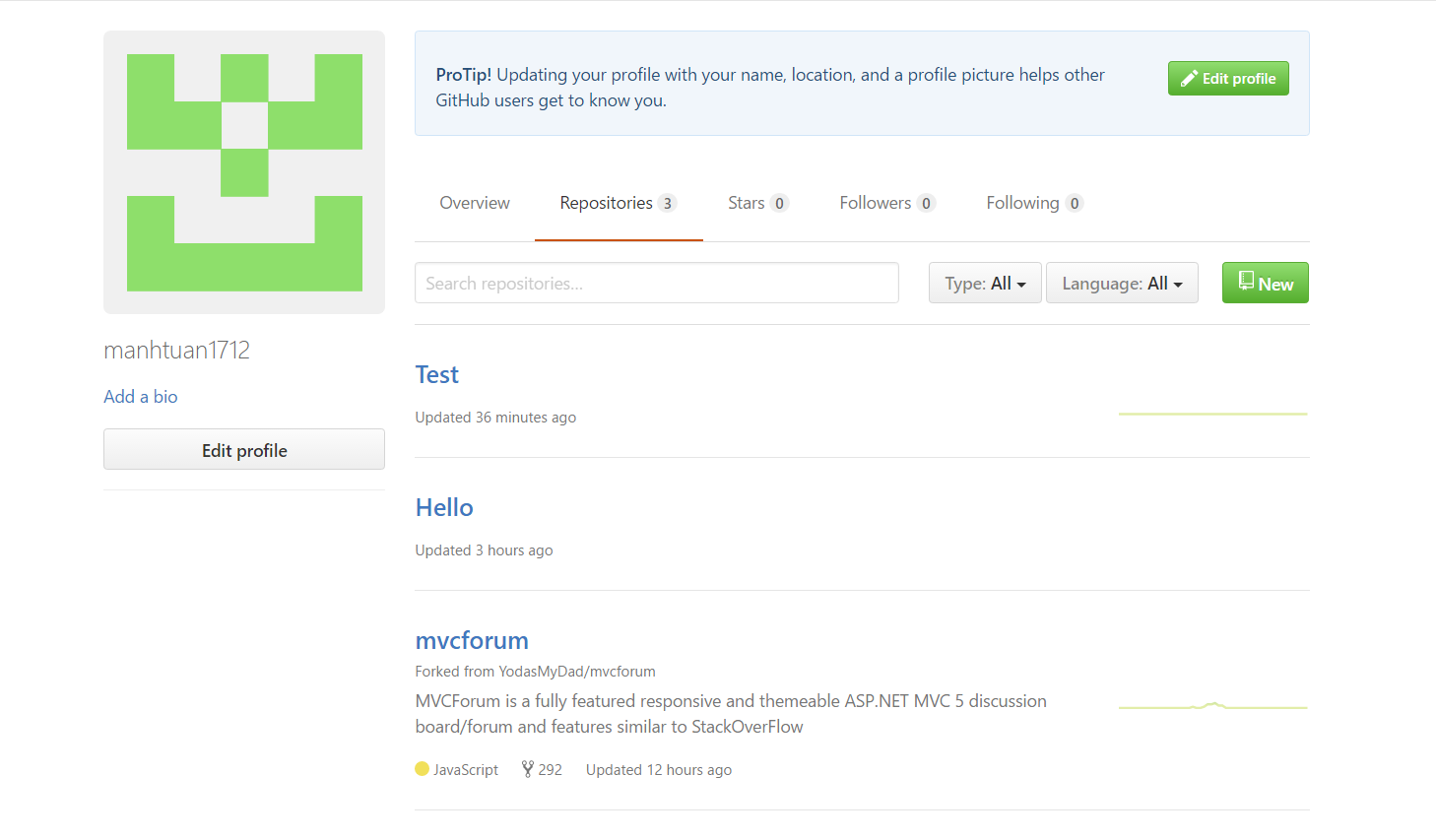
****

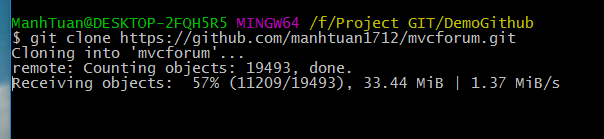
****

****

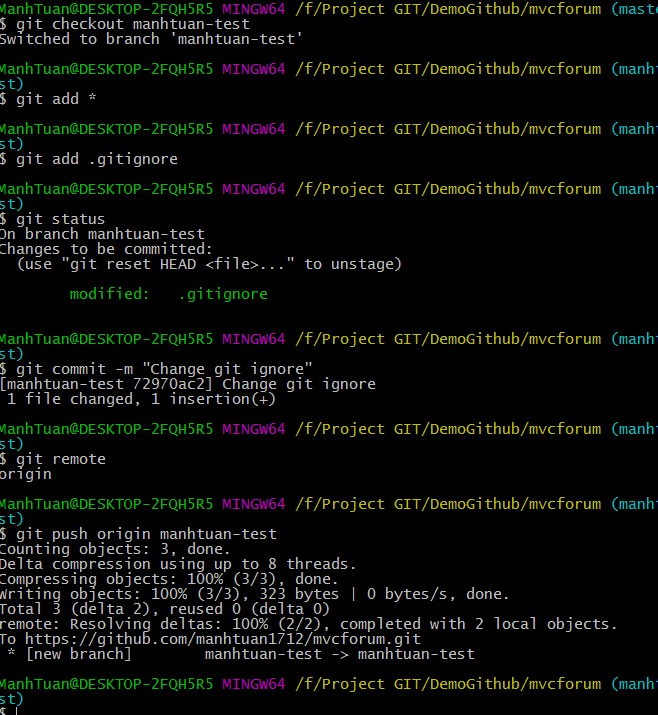
****

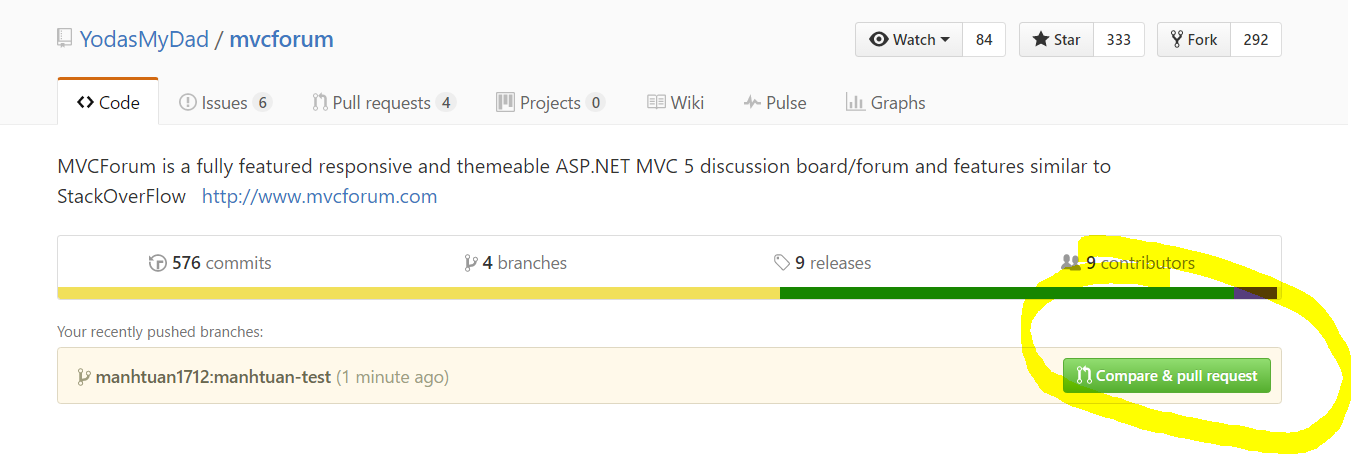
**Đóng góp 1 dự án GitHub**

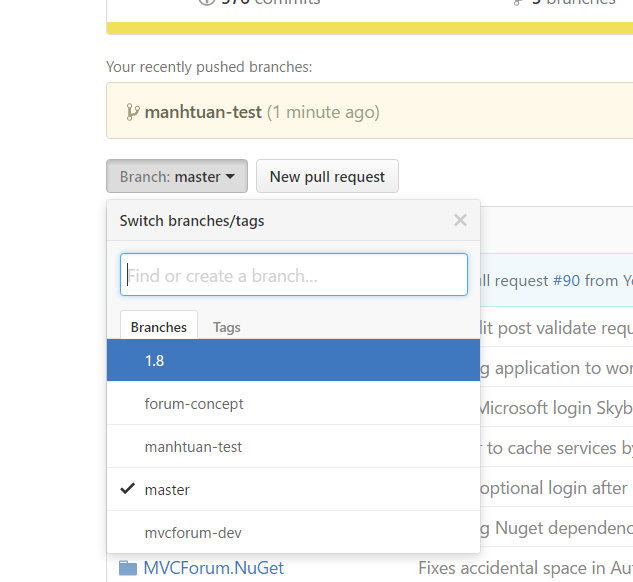
** **

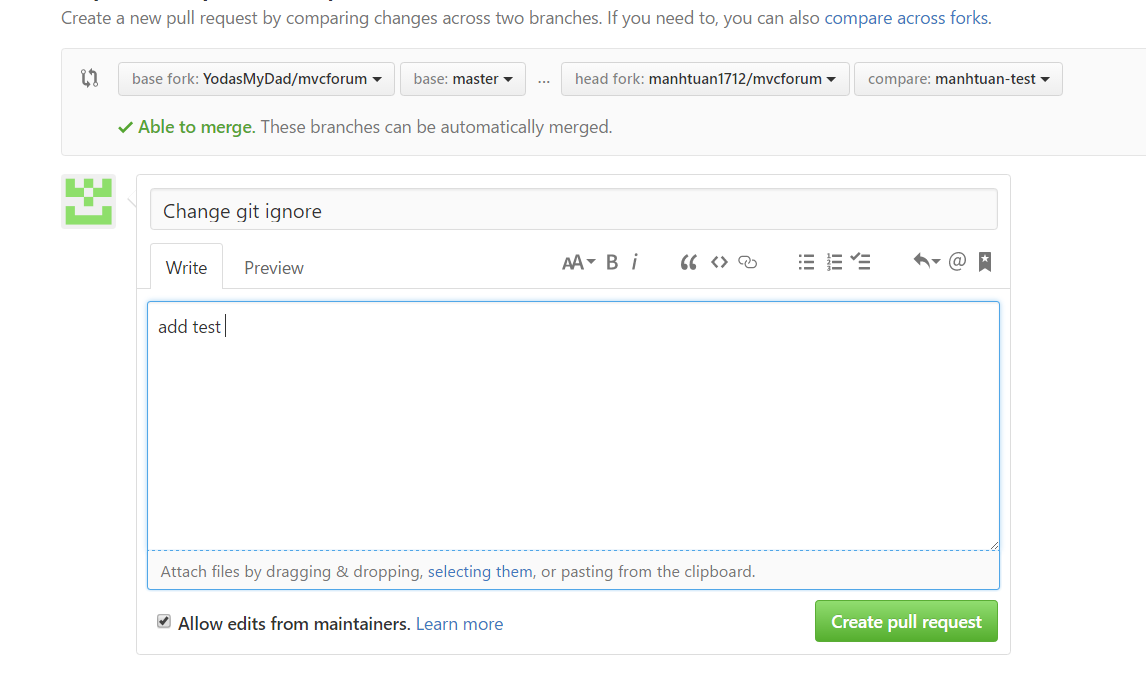
****

****

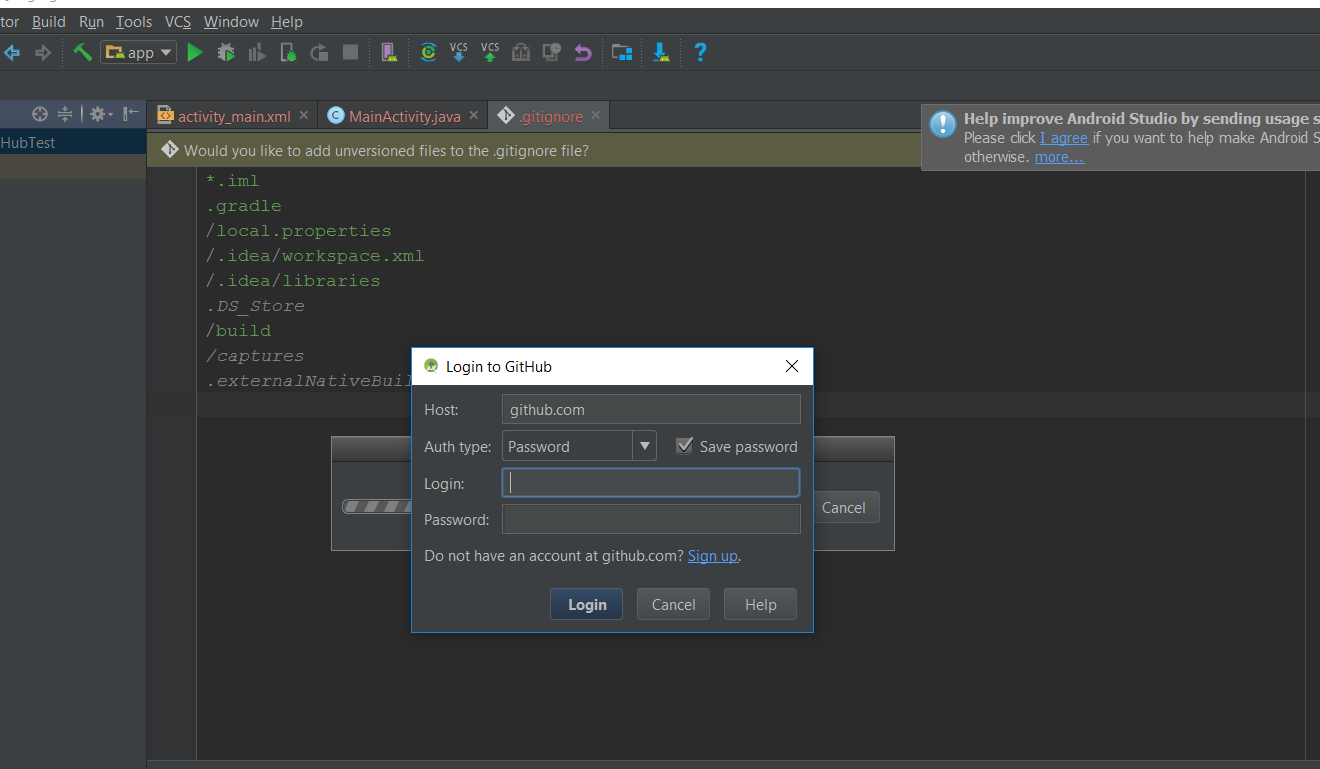
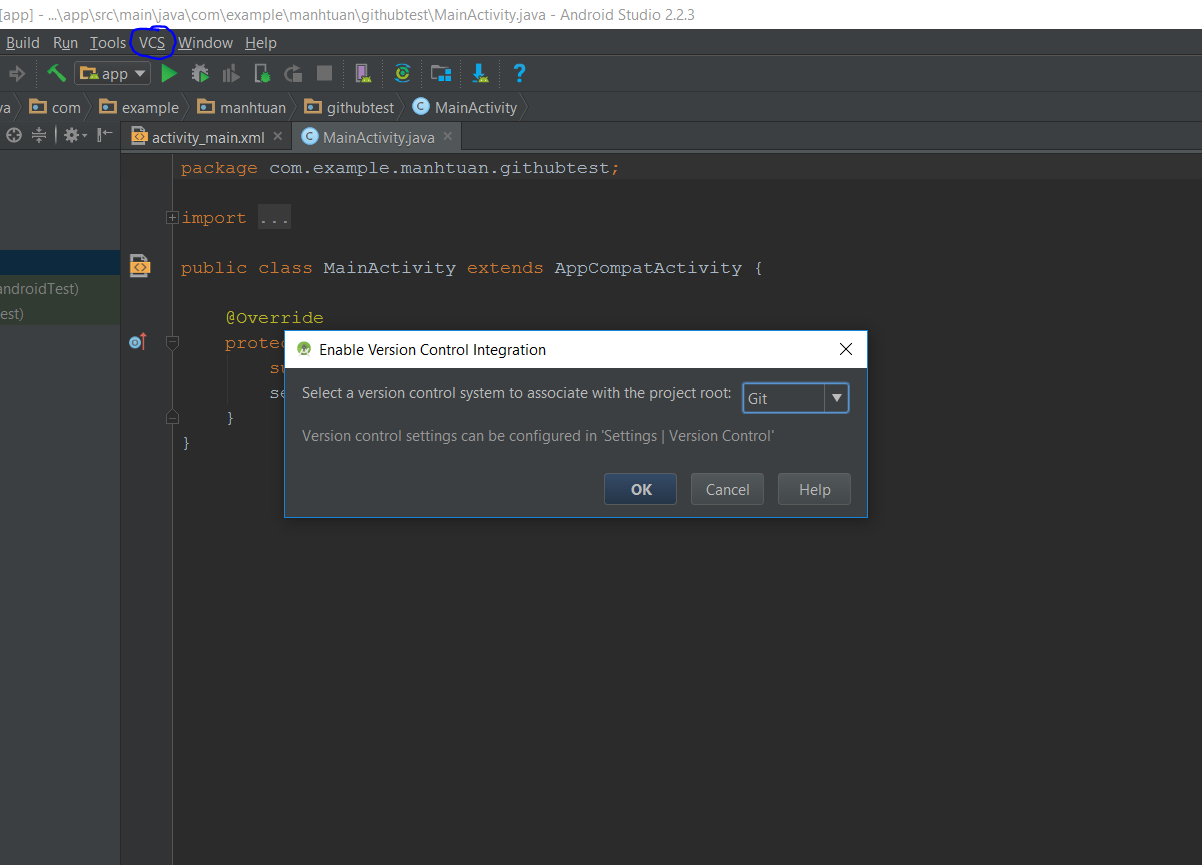
****

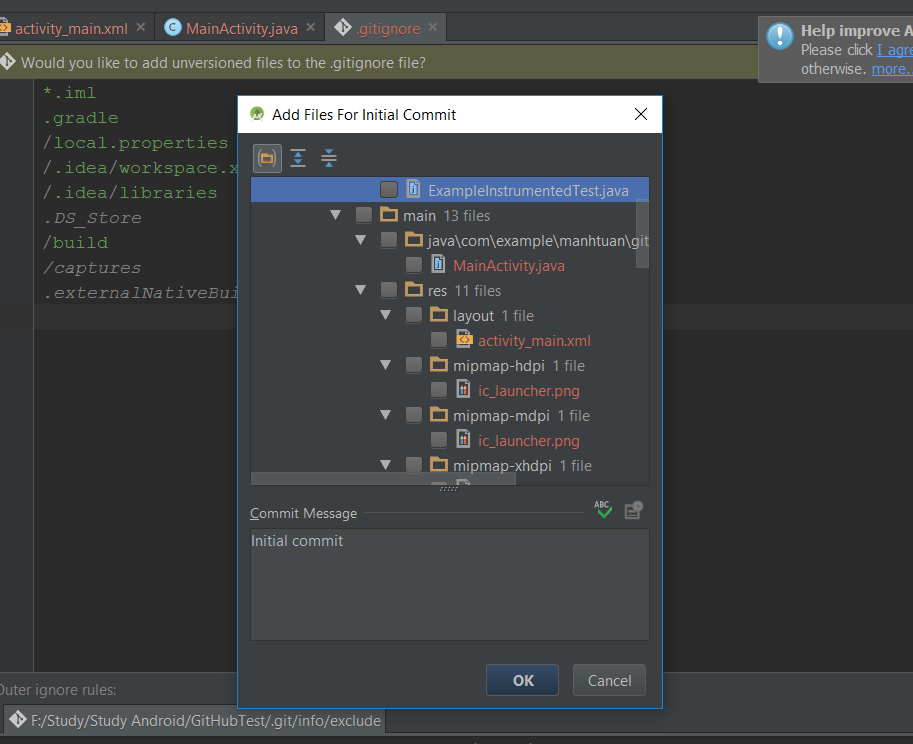
****

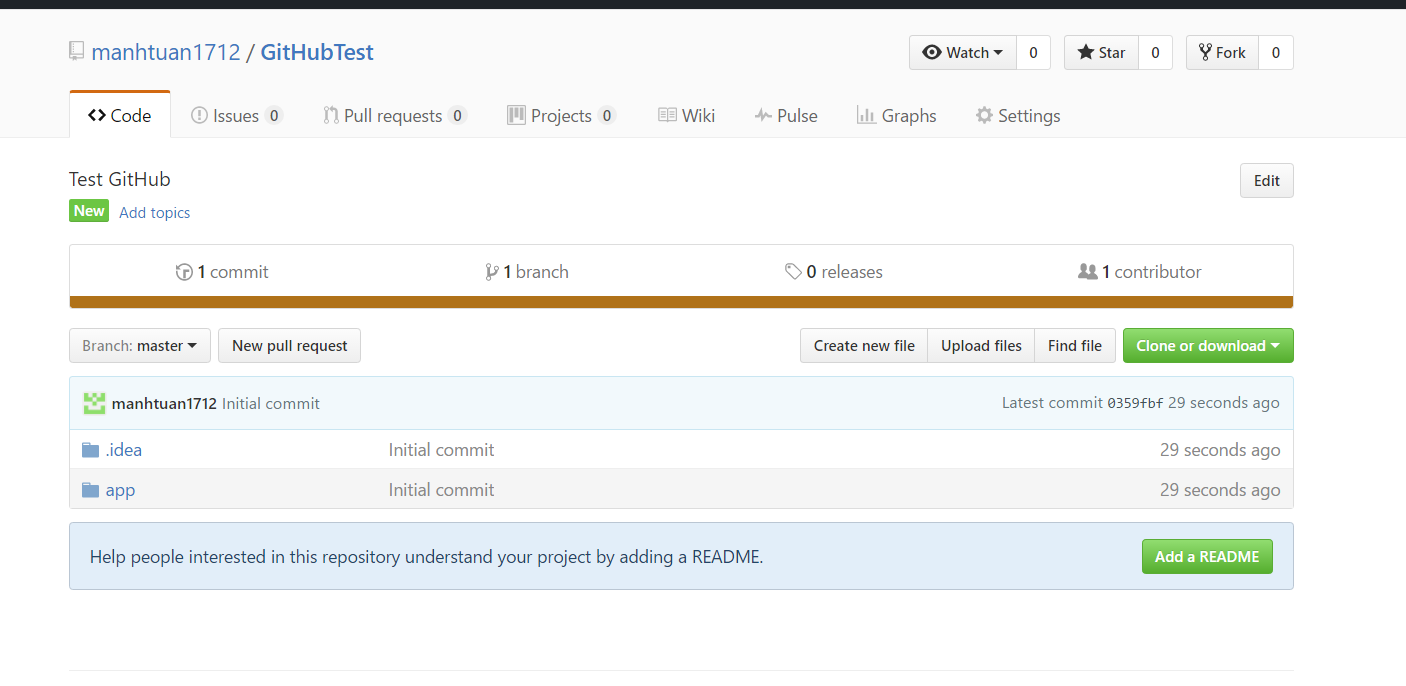
****

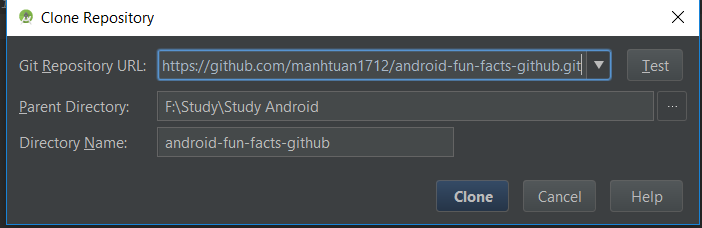
****

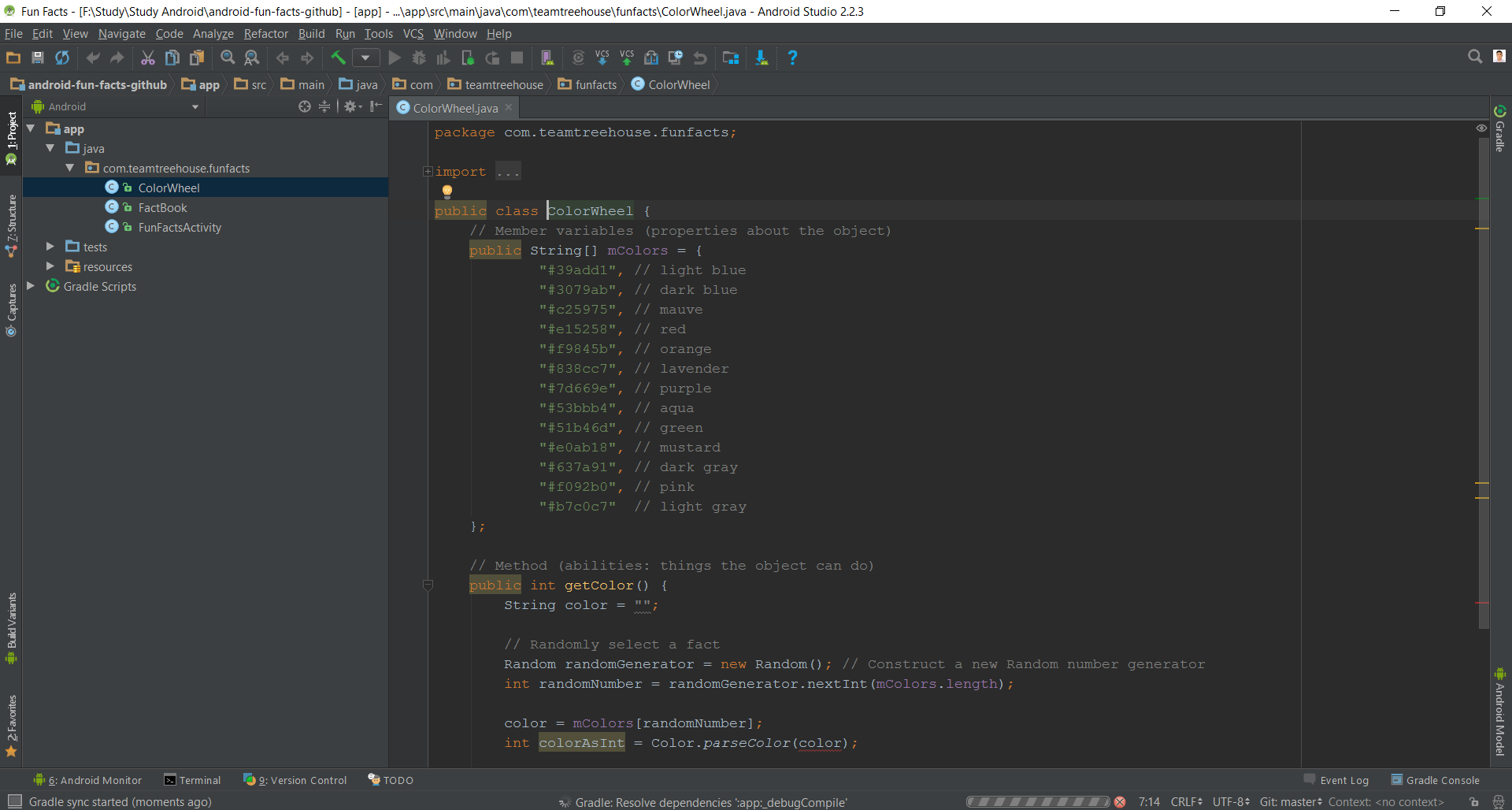
**Github và android studio**

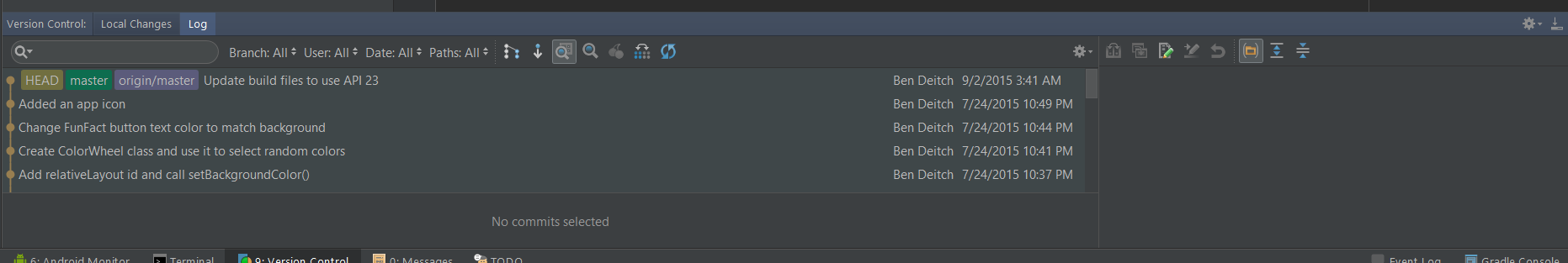
****

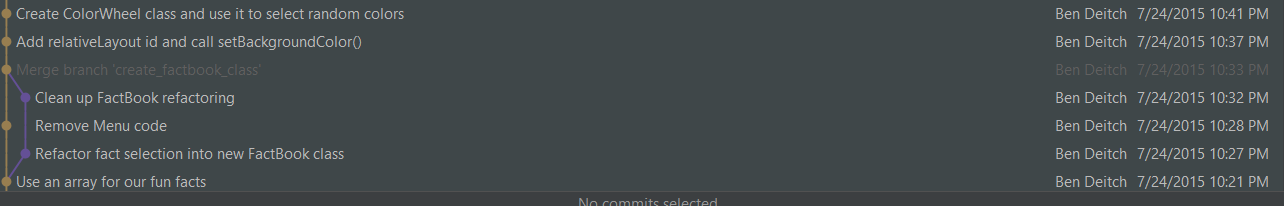
****

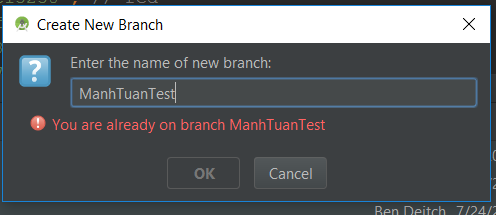
****

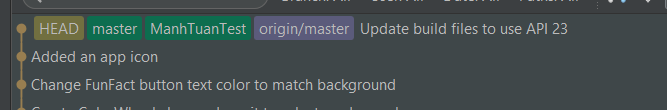
****

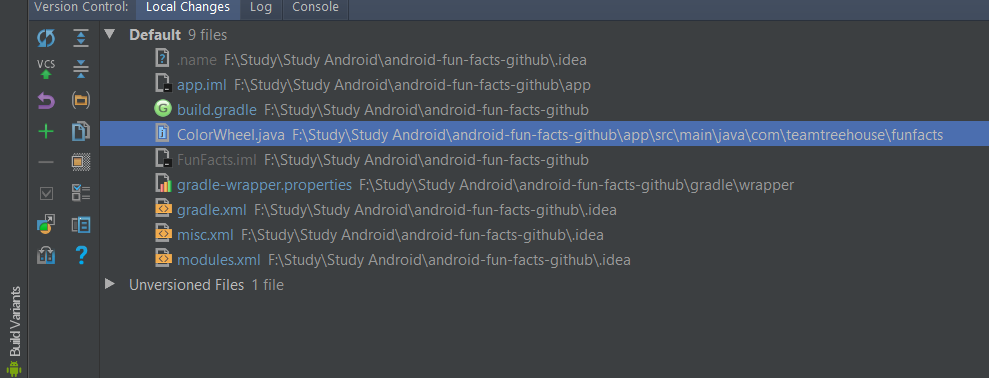
****

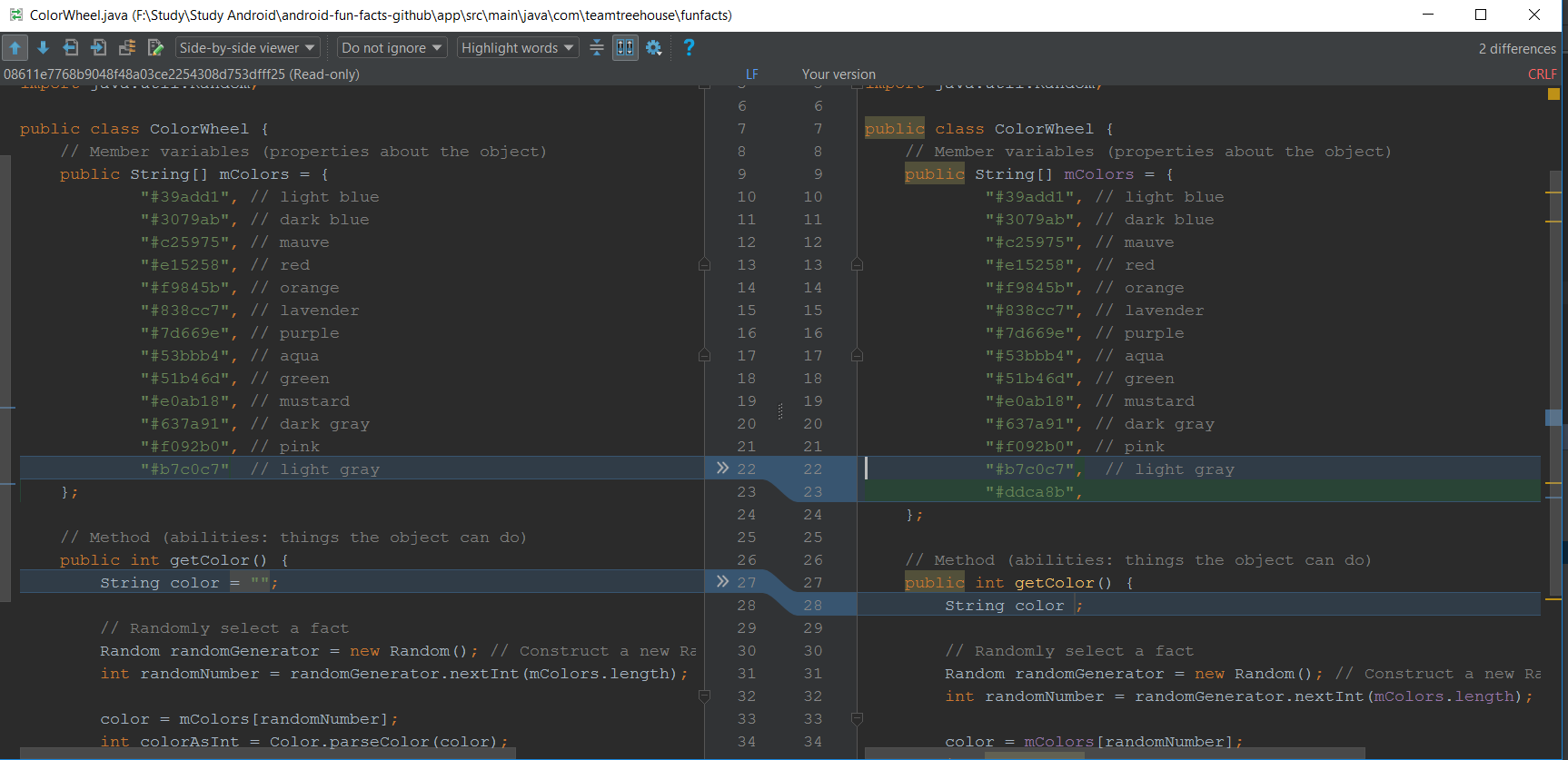
****

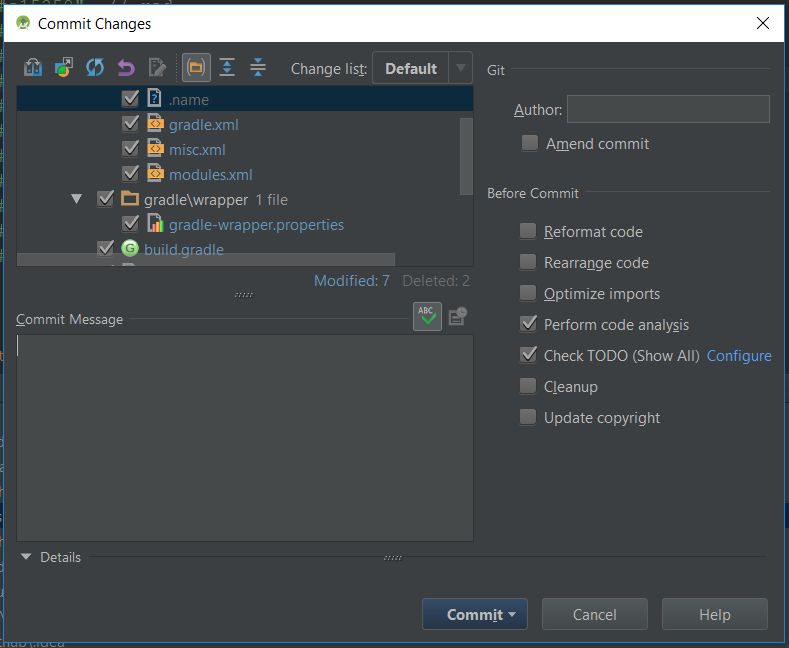
****

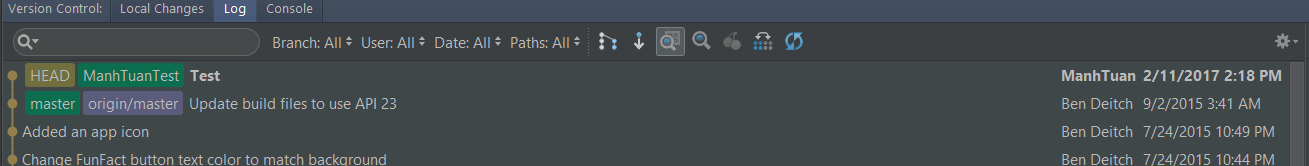
****

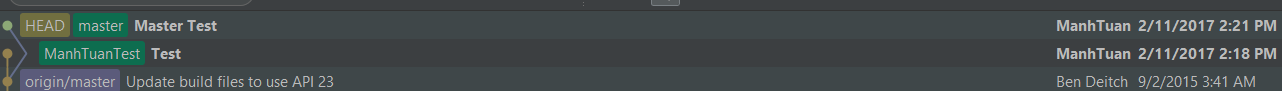
****

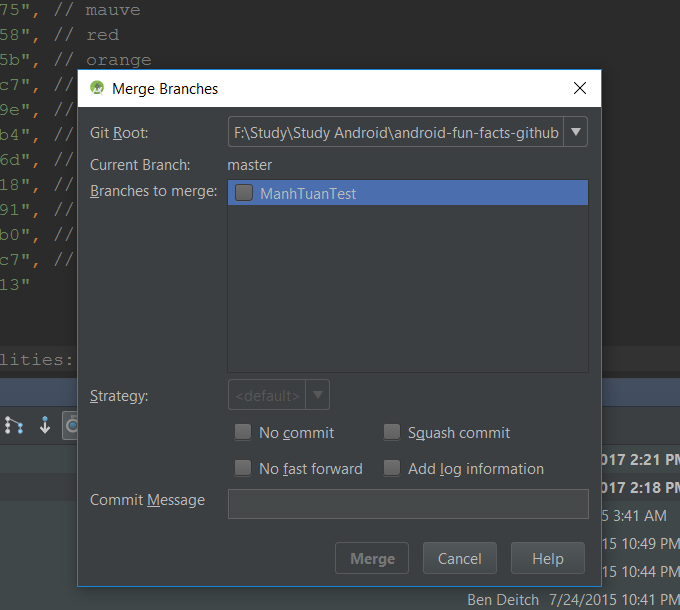
****

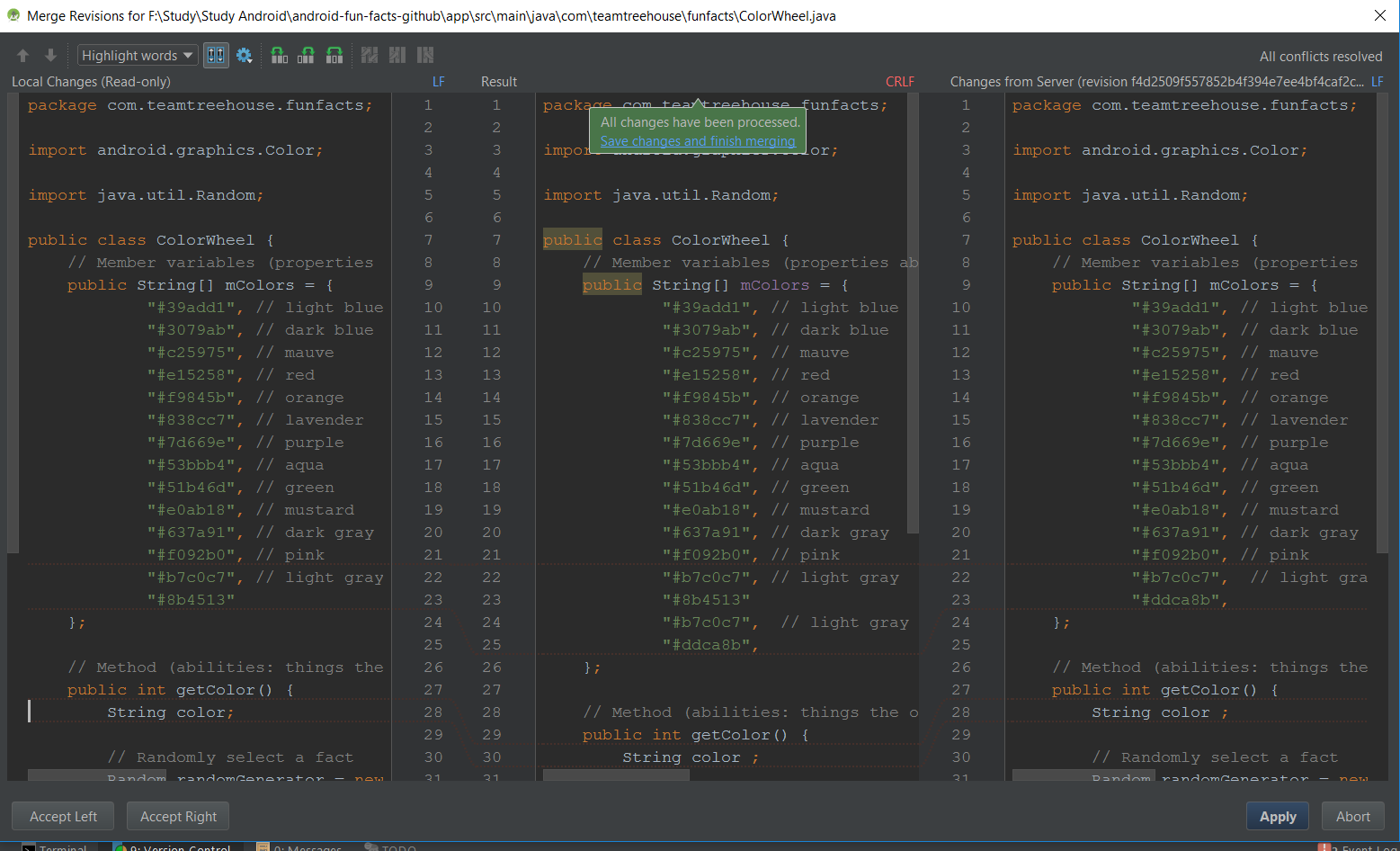
****

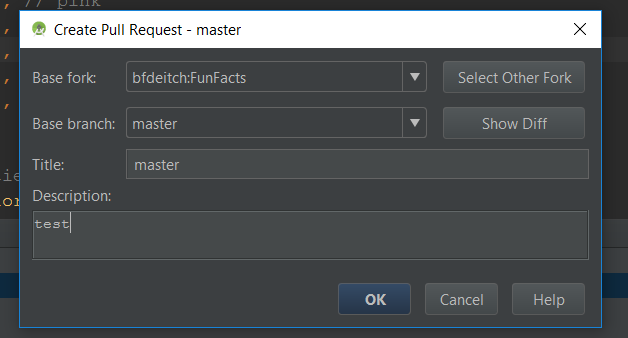
****

****

****

****

****

****