

HỆ THỐNG QUẢN LÝ SỰ CỐ JIRA VÀ QUẢN LÝ PHIÊN BẢN PHÂN TÁN – GIT



git

JIRA & Git

- Bạn sẽ được làm quen với hệ thống JIRA, Git và cách sử dụng
- Hiểu được các khái niệm và thuật ngữ trong Git như Clone, Branch, Stage, Commit, Tracking, Merge, Pull, Push, Reset...và nó sẽ là câu trả lời cho bạn để hiểu Git là cái gì và cách sử dụng nó, tại sao nên sử dụng nó.

NỘI DUNG

CÔNG CỤ CHUẨN BỊ	2
CÁC THUẬT NGỮ	2
LÀM VIỆC VỚI JIRA	4
Tạo tài khoản JIRA và liên kết với Source control	4
Liên kết JIRA và Bitbucket:	6
Từ Bitbucket	6
Từ Bảng quản trị của JIRA	10
Tạo JIRA Issue	11
GIT LÀ GÌ	14
REMOTE GIT REPOSITORY	14
CẤU HÌNH	15
Dùng dòng lệnh:	15
Dùng TortoiseGit	16
Dùng Sourcetree	18
LÀM VIỆC VỚI GIT	19
Tạo Local Repository	19
Kết nối với Remote Repository	19
Thêm mới một (hoặc nhiều) remote repository:	19
Thay đổi url của kho lưu trữ từ xa hiện tại	19
Sao chép (Clone) một Remote Repository	20
Add và Commit	20
Smart Commit	20
Git push - Đưa thay đổi mã nguồn lên remote repository	21
Git Log – Xem lại nhật kí của các commit	21
Tạo nhánh	22
Tạo pull request/ merge nhánh vào nhánh chính (master)	23
Git merge	24
Tạo pull request	24
Conflict – Xung đột và cách xử lí	28
Tự động Merge thành công	29
Tự động Merge không thành công	29
Revert commit	31

CÔNG CỤ CHUẨN BỊ

Tải công cụ git cho windows: <https://git-scm.com/download/win>

Tài khoản tại trang quản lí source code và quản lí sự cố (ở đây sử dụng JIRA):

- Bitbucket: <https://bitbucket.org/>
- JIRA: <https://www.atlassian.com/software/jira/try>

Ngoài ra, ta có thể cài đặt thêm các công cụ hỗ trợ giao diện đồ họa sử dụng song song với màn hình dòng lệnh của Git-SCM:

- TortoiseGit: <https://tortoisegit.org>
- Sourcetree
- Folk

CÁC THUẬT NGỮ

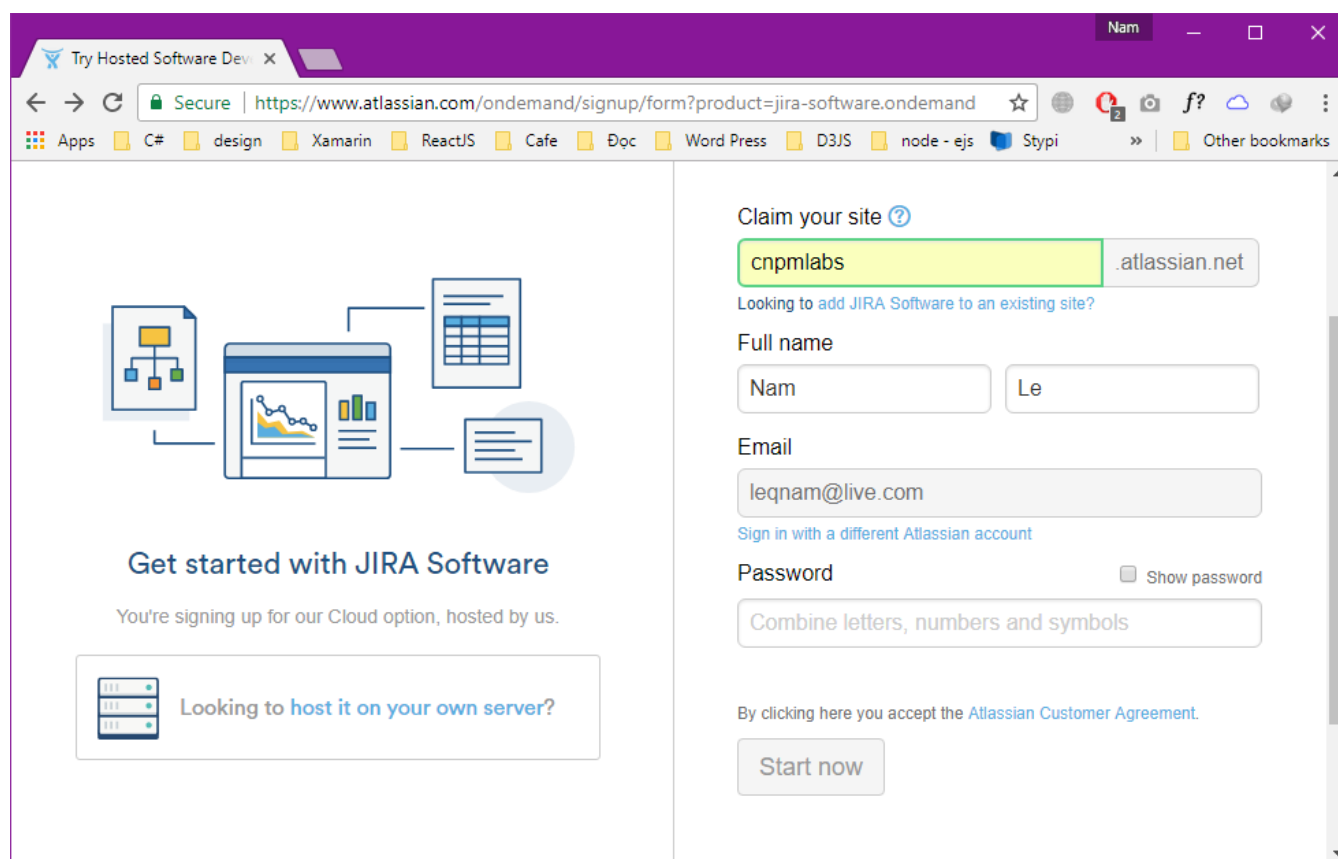
Git		
SCM		
Working Copy		
Branch		
Commit		
Push		
Stash		
Revert		
JIRA Issue		

JIRA Key		
----------	--	--

LÀM VIỆC VỚI JIRA

Tạo tài khoản JIRA và liên kết với Source control

Truy cập địa chỉ: <https://www.atlassian.com/software/jira/try> , chọn plan cần sử dụng, trong trường hợp này, ta chọn dùng thử 7 ngày, sau thời gian này cần bỏ ra 10\$ mỗi tháng để duy trì JIRA, điền các thông tin vào cho dự án:



Try Hosted Software Dev X

Secure | <https://www.atlassian.com/ondemand/signup/form?product=jira-software.ondemand>

Apps C# design Xamarin ReactJS Cafe Đọc Word Press D3JS node - ejs Styti Other bookmarks

Get started with JIRA Software

You're signing up for our Cloud option, hosted by us.

☐ Looking to host it on your own server?

Claim your site ?

cnplabs .atlassian.net

Looking to add JIRA Software to an existing site?

Full name

Nam Le

Email

leqnam@live.com

[Sign in with a different Atlassian account](#)

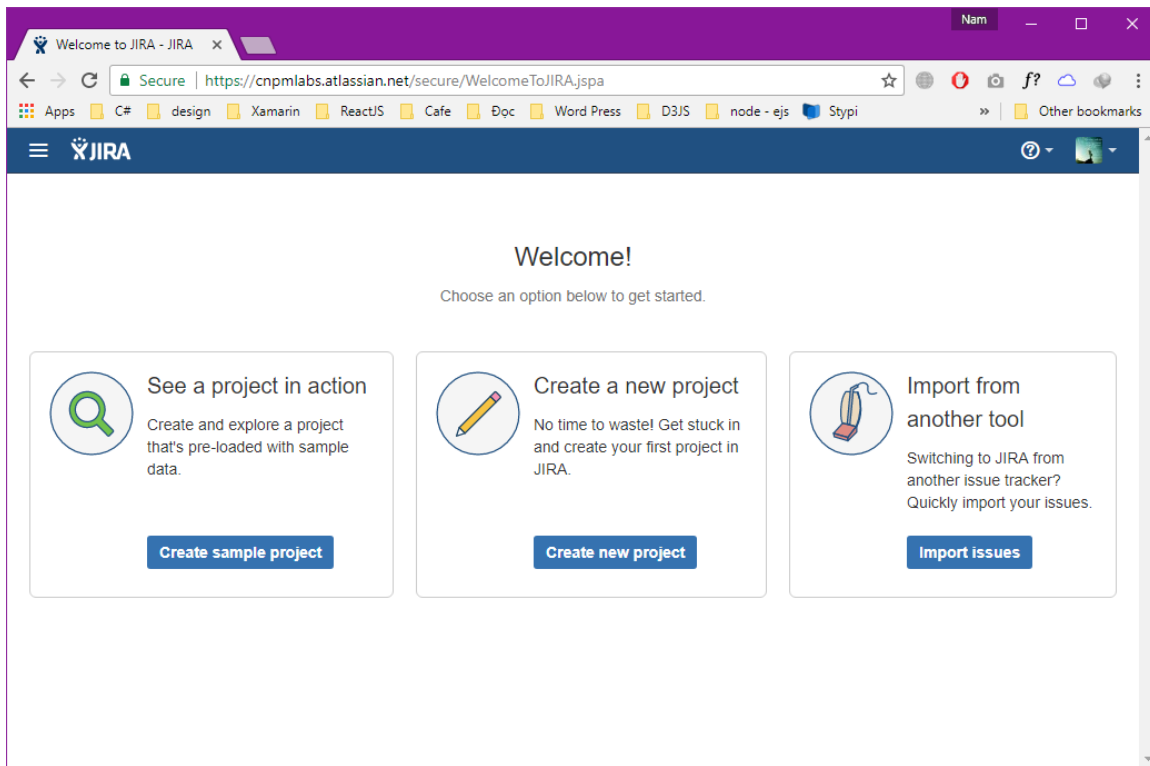
Password ☐ Show password

Combine letters, numbers and symbols

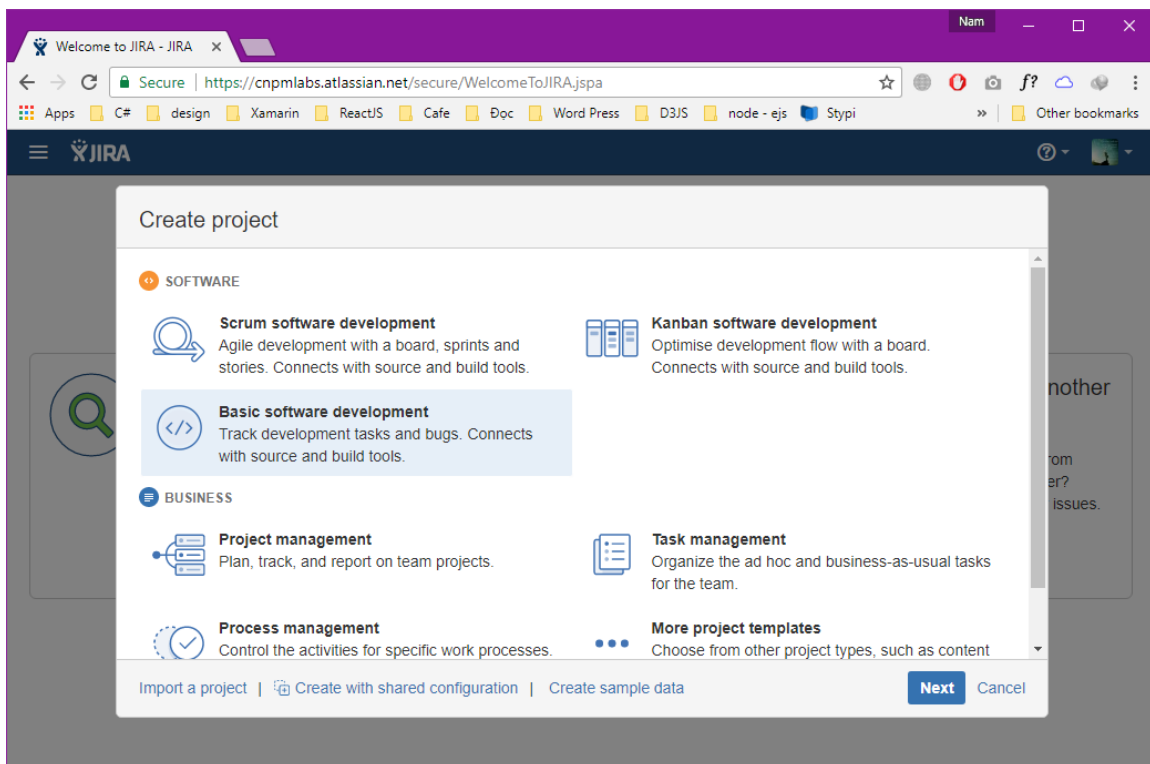
By clicking here you accept the [Atlassian Customer Agreement](#).

Start now

Tiếp đến sẽ là màn hình cài đặt trong giây lát, Sau khi tạo xong, ta tiến hành cấu hình JIRA để nó có thể liên kết được với Source Control, JIRA chỉ hỗ trợ liên kết đến Souce Control là Bitbucket và Github. Ta tạo một project cần quản lí:



Chọn mô hình project là Scrum hoặc Kanban, ta chỉ chọn Basic Software Development



Bước tiếp theo là thiết lập tên, key cho project. Key sẽ dùng để chỉ định một git commit liên kết đến một JIRA issue cụ thể.

Welcome to JIRA - JIRA

Secure | <https://cnplabs.atlassian.net/secure/WelcomeToJIRA.jspa>

Apps C# design Xamarin ReactJS Cafe Đọc Word Press D3JS node - ejS Stypi Other bookmarks

JIRA

Welcome!

Basic software development

Name Max. 80 characters.

Key Max. 10 characters.

[Basic software development](#)
Specify a descriptive name and key for your project. For example, the name of the application that you are developing.

You also need to choose a project lead, if you have more than one user. This should be the person who manages development for this project.

Back Submit Cancel

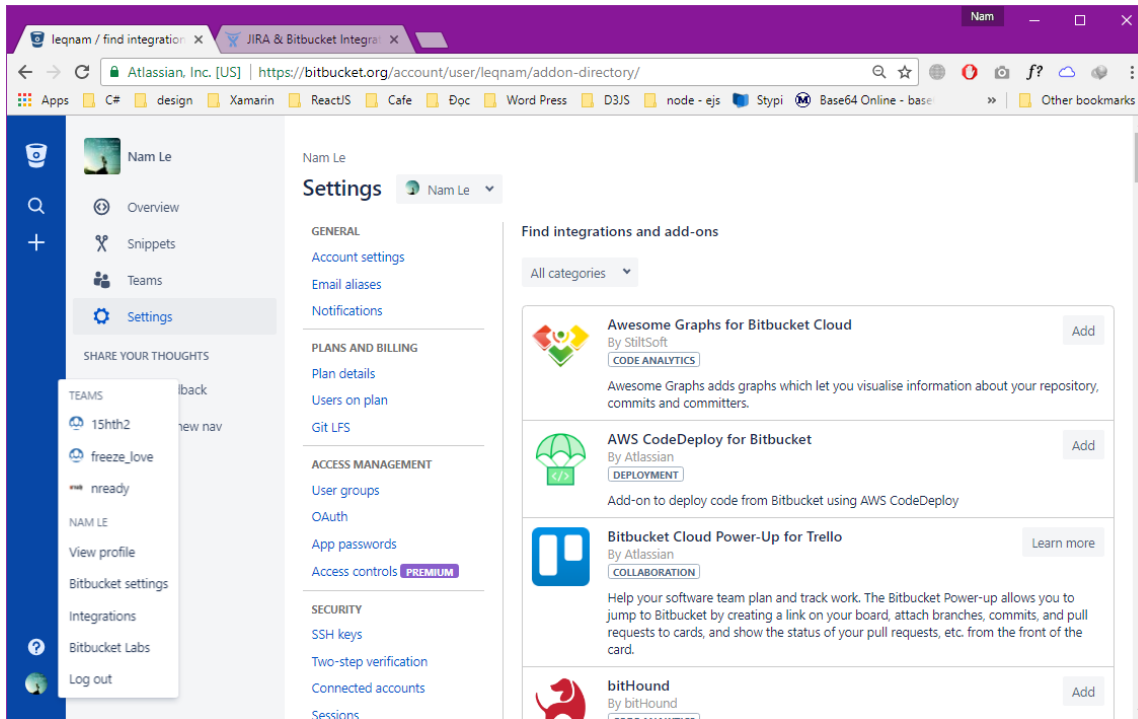
Liên kết JIRA và Bitbucket:

Ta tiến hành liên kết JIRA và Bitbucket với nhau để dễ dàng trong việc quản lý các Issue và các Git Commit.

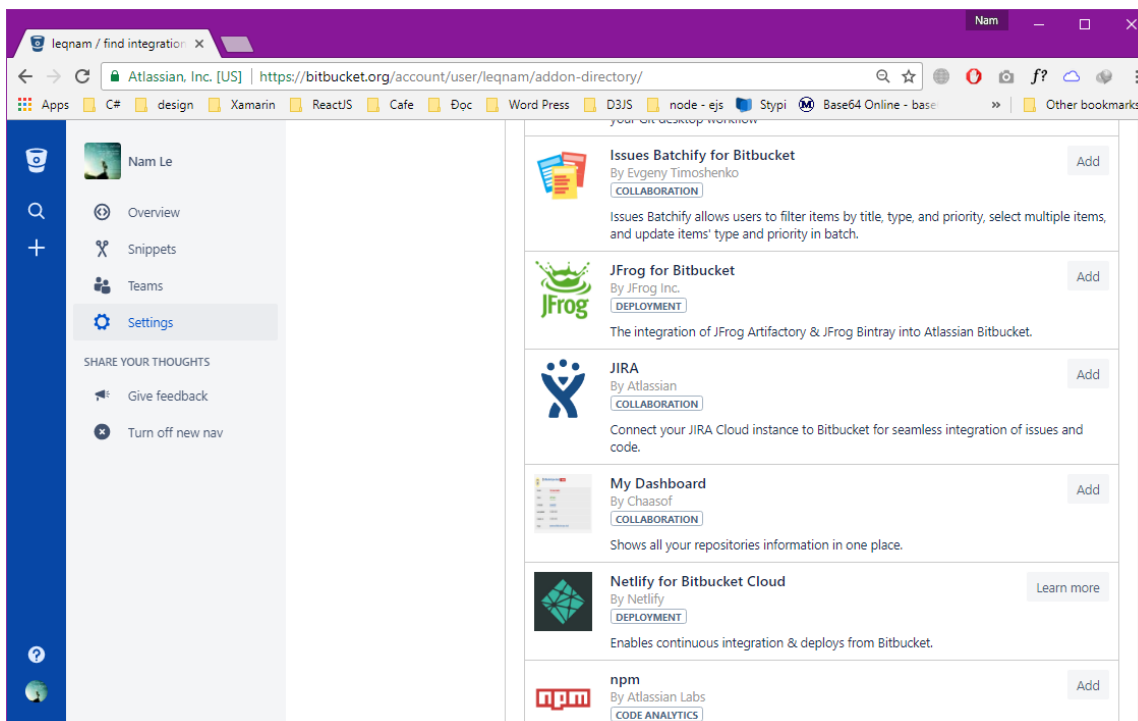
Có hai hướng thực hiện, một là từ Bitbucket, hai là từ bảng quản trị của JIRA.

Từ Bitbucket

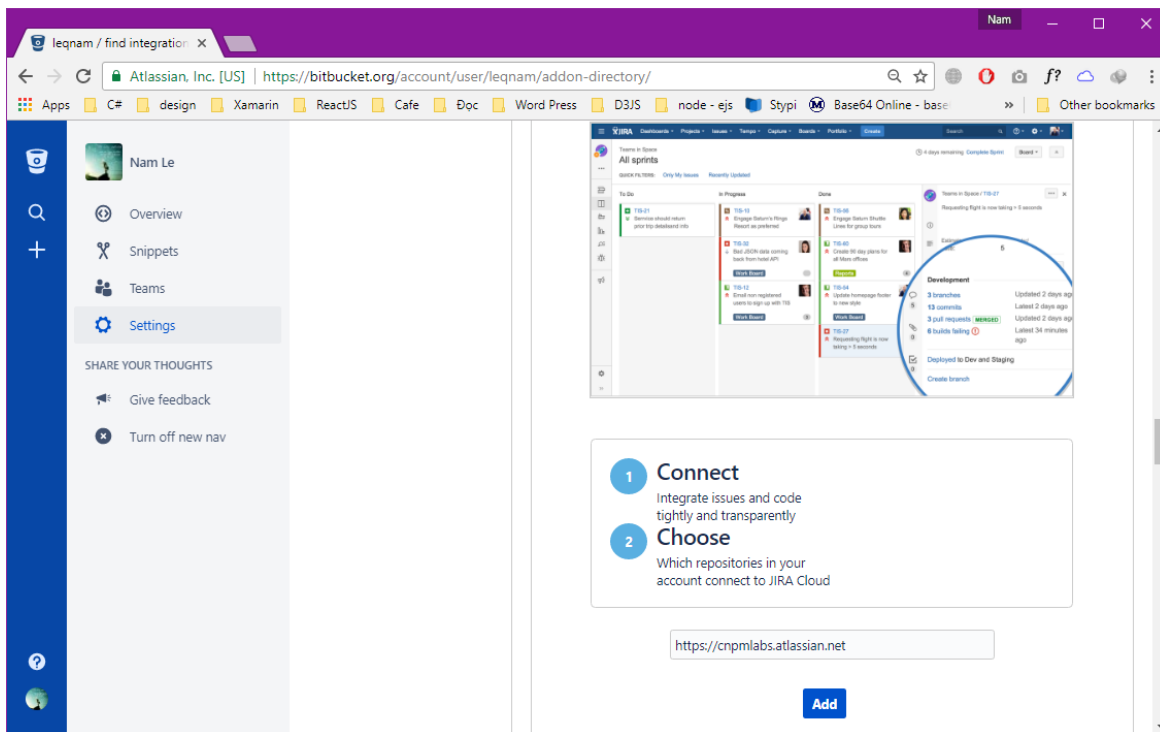
Chọn Bitbucket Setting:



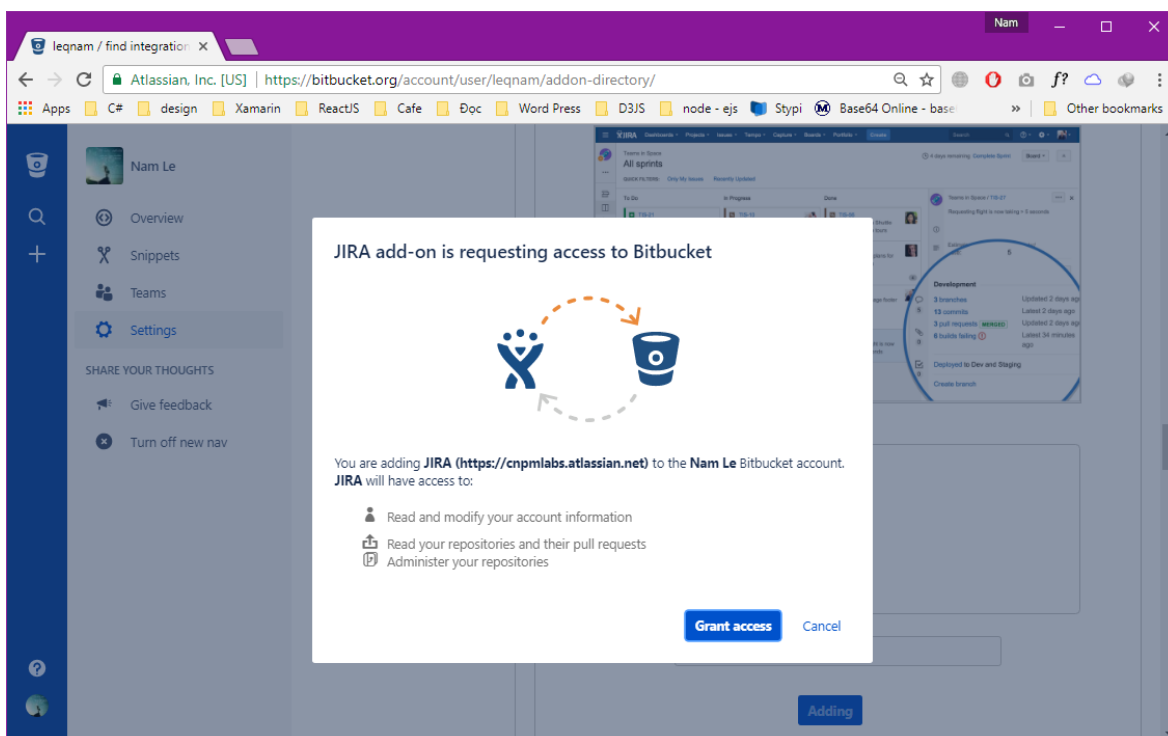
Chọn JIRA là add-on.



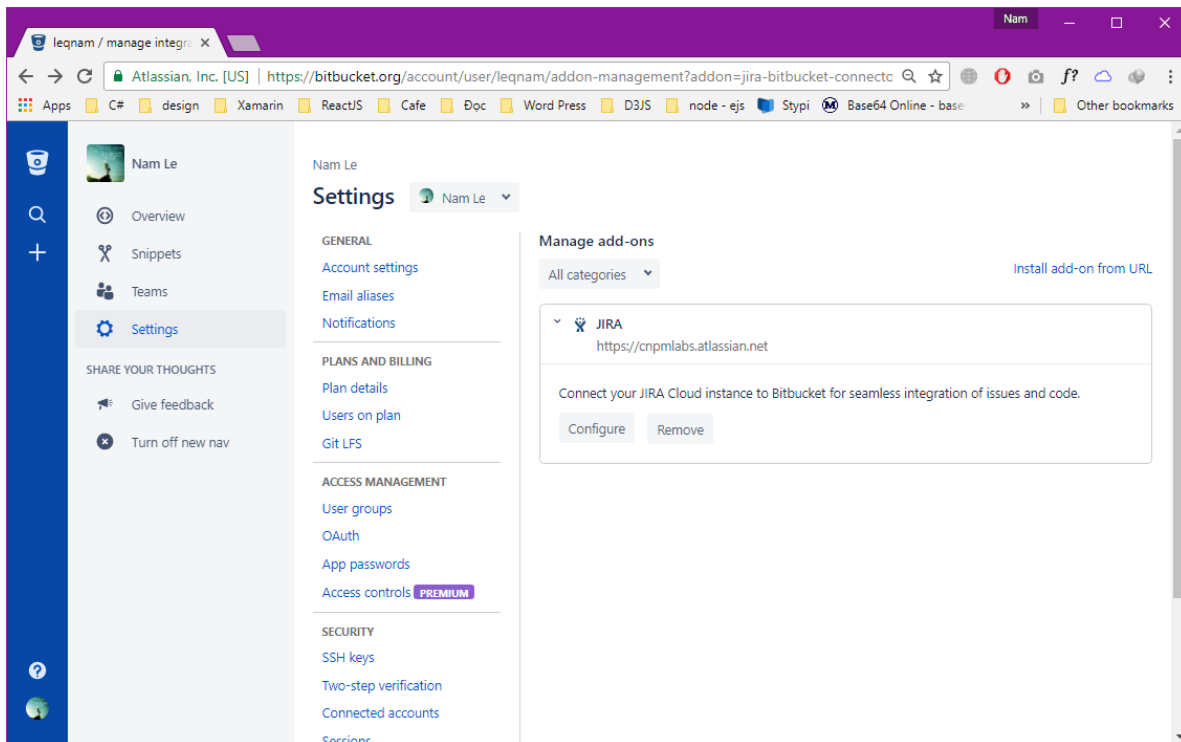
Tiến hành thêm liên kết repository Bitbucket vào:



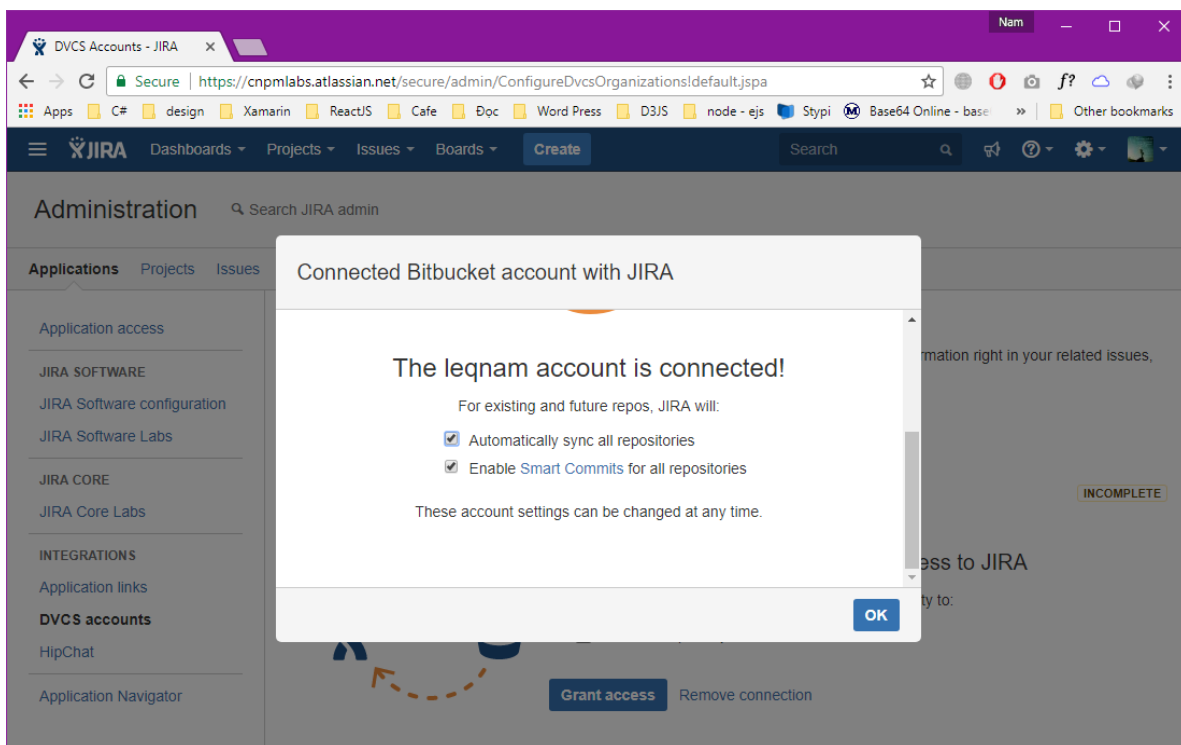
Tiến hành cấp quyền cho JIRA có thể truy cập:



Sau khi hoàn tất sẽ như hình sau:



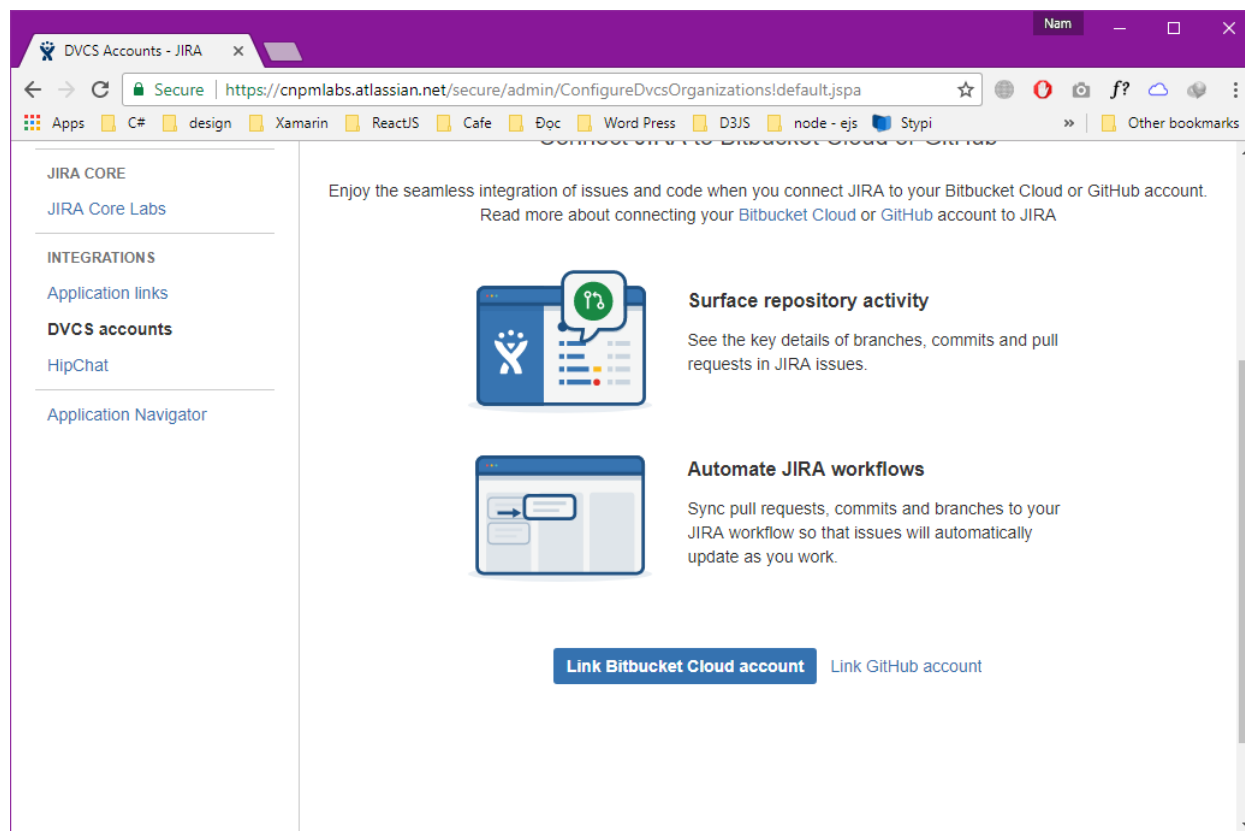
Tuy nhiên, cần phải vào JIRA Dashboard Administrator để tiến hành xác nhận quyền truy cập:



Sau khi bấm Ok xác nhận, tài khoản Bitbucket đã được liên kết với JIRA.

Từ Bảng quản trị của JIRA

Từ bảng quản trị của JIRA, chọn DVCS accounts, chọn tiếp “Link Bitbucket Cloud Account” (hoặc Link Github Account)



Và sau đó là thao tác y hệt như cách 1, gồm thêm repository và các bước xác nhận quyền truy cập.

Tạo JIRA Issue

Tiến hành tạo mới một Issue bằng việc nhấn vào Create.

Thiết đặt cho issue cần tạo, bao gồm các thông tin chính như sau:

- **Project:** Chọn tên của project cần tạo một Issue
- **Issue type:** mặc định có 4 loại, New Feature, Task, Bug, Epic
- **Summary:** Nội dung ngắn gọn cho Issue, thông thường ta ngầm định nó là tiêu đề
- **Description:** Mô tả nội dung cho Issue
- **Priority:** Mức độ ưu tiên, Highest/High/Low/Lowest/Medium
- **Labels:** Nhãn cho Issue, dùng khi kết hợp tìm kiếm và filter theo loại dùng Query.
- **Original Estimate:** Thời gian hoàn thành một Issue, đơn vị tính w/ d/ h /m /s
- **Remaining Estimate:** Thời gian còn lại của Issue, đơn vị tính w/ d/ h /m /s
- **Attachment:** Các tệp đính kèm nếu có
- **Due Date:** Ngày thực hiện
- **Linked Issues:** Có các loại như blocks/is blocked by/clones/is cloned by/duplicates/is duplicated by/relates to
- **Issue:** Issue liên quan đến Issue đang tạo
- **Assignee:** Người sẽ đảm nhận Issue này, nếu chưa xác định ai thì để trống, nếu là chính mình thì Assign to me, hoặc nhập tên của người cần đảm nhận vào.
- **Epic Link:** Nếu Issue cần liên hệ với một Epic thì chọn Epic.

The screenshot shows the JIRA 'Create issue' dialog box. The browser address bar shows the URL 'https://cnpmmlabs.atlassian.net/projects/CNPM/issues/?filter=allopenissues'. The JIRA navigation bar at the top includes 'Dashboards', 'Projects', 'Issues', 'Boards', and a 'Create' button. The dialog box has a title 'Create issue' and a 'Configure fields' button. The 'Project' dropdown is set to 'cnpmmlabs (CNPM)'. The 'Issue Type' dropdown is set to 'Task'. Below the 'Issue Type' dropdown, a message states: 'Some issue types are unavailable due to incompatible field configuration and/or workflow associations.' The 'Summary' field contains the text 'Tạo tài khoản GIT và SVN'. The 'Component/s' field is set to 'None'. The 'Description' field contains the text 'Tạo tài khoản GIT tại trang https://bitbucket.org/' and 'Tạo tài khoản SVN tại trang https://www.assembla.com/'. At the bottom right, there are buttons for 'Create another', 'Create', and 'Cancel'.

GIT LÀ GÌ

Git là tên gọi của một **Hệ thống quản lý phiên bản phân tán** (*Distributed Version Control System – DVCS*) là một trong những hệ thống quản lý phiên bản phân tán phổ biến nhất hiện nay. DVCS nghĩa là hệ thống giúp mỗi máy tính có thể lưu trữ nhiều phiên bản khác nhau của một mã nguồn được nhân bản (**clone**) từ một kho chứa mã nguồn (**repository**), mỗi thay đổi vào mã nguồn trên máy tính sẽ có thể ủy thác (**commit**) rồi đưa lên máy chủ nơi đặt kho chứa chính. Và một máy tính khác (nếu họ có quyền truy cập) cũng có thể clone lại mã nguồn từ kho chứa hoặc clone lại một tập hợp các thay đổi mới nhất trên máy tính kia. Trong Git, thư mục làm việc trên máy tính gọi là **Working Tree**.

Cơ chế lưu trữ phiên bản của Git là nó sẽ tạo ra một “ảnh chụp” (*snapshot*) trên mỗi tập tin và thư mục sau khi commit, từ đó nó có thể cho phép bạn tái sử dụng lại một ảnh chụp nào đó mà bạn có thể hiểu đó là một phiên bản. Đây cũng chính là lợi thế của Git so với các DVCS khác khi nó không “lưu cứng” dữ liệu mà sẽ lưu với dạng snapshot.

REMOTE GIT REPOSITORY

Để làm việc với Git và tổ chức quản lý mã nguồn (source-code) thì ta cần có một Remote Git Repository đóng vai trò là 1 server lưu trữ mã nguồn.

Git có 4 loại giao thức (protocols) để trao đổi dữ liệu: **Local**, **Secure Shell (SSH)**, **Git**, và **HTTPS**.

CẤU HÌNH

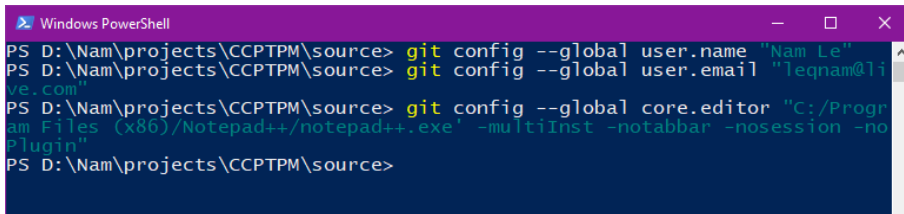
Sau khi cài đặt Git-SCM xong, cần phải thiết lập các biến môi trường. Thông thường là thiết lập username, email cho các commit.

Sau khi cài đặt xong Git-SCM, ta tiến hành cấu hình các thông tin như tên người dùng, email, các công cụ merge tool, diff tool, editor.

Cấu hình tên người dùng, email, core-editor, trên Windows, các thiết lập sẽ được lưu tại "C:\Users\<user>\.gitconfig", với <user> là tên người dùng hiện tại đăng nhập vào hệ thống.

Các cách cấu hình chính:

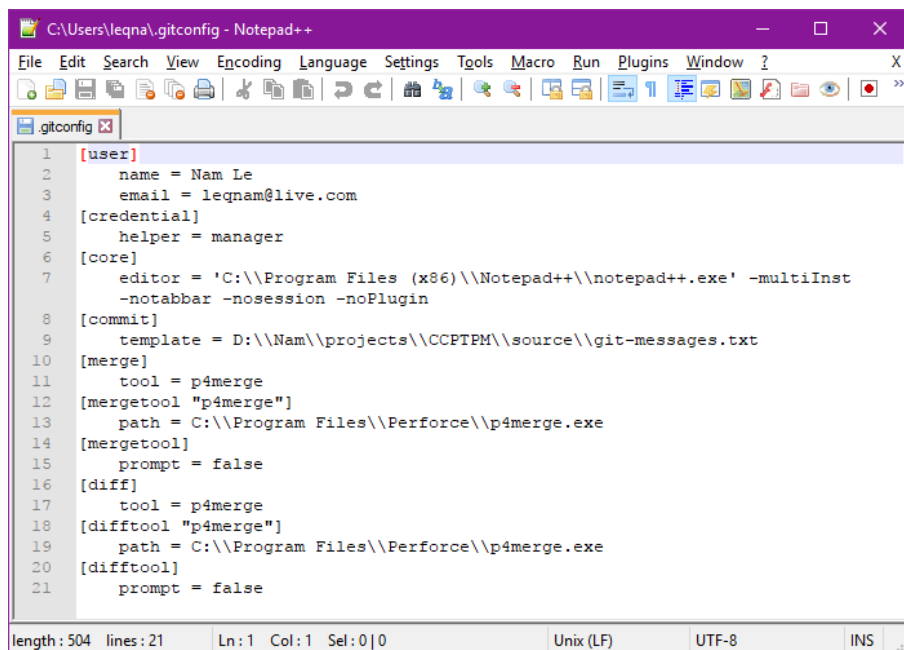
Dùng dòng lệnh:



```

PS D:\Nam\projects\CCPTPM\source> git config --global user.name "Nam Le"
PS D:\Nam\projects\CCPTPM\source> git config --global user.email "leqnam@live.com"
PS D:\Nam\projects\CCPTPM\source> git config --global core.editor "C:/Program Files (x86)/Notepad++/notepad++.exe -multiInst -notabbar -nosession -noPlugin"
PS D:\Nam\projects\CCPTPM\source>
  
```

Hoặc mở file **.config** bằng chương trình chỉnh sửa như Notepad, Visual Code,... như trong hình sau:

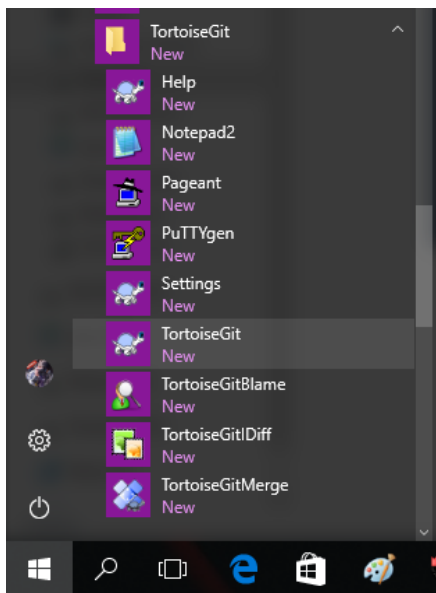


```

1 [user]
2   name = Nam Le
3   email = leqnam@live.com
4 [credential]
5   helper = manager
6 [core]
7   editor = 'C:\\Program Files (x86)\\Notepad++\\notepad++.exe' -multiInst
8             -notabbar -nosession -noPlugin
9 [commit]
10  template = D:\\Nam\\projects\\CCPTPM\\source\\git-messages.txt
11 [merge]
12  tool = p4merge
13 [mergetool "p4merge"]
14   path = C:\\Program Files\\Perforce\\p4merge.exe
15 [mergetool]
16  prompt = false
17 [diff]
18  tool = p4merge
19 [difftool "p4merge"]
20   path = C:\\Program Files\\Perforce\\p4merge.exe
21 [difftool]
22  prompt = false
  
```


Dùng TortoiseGit

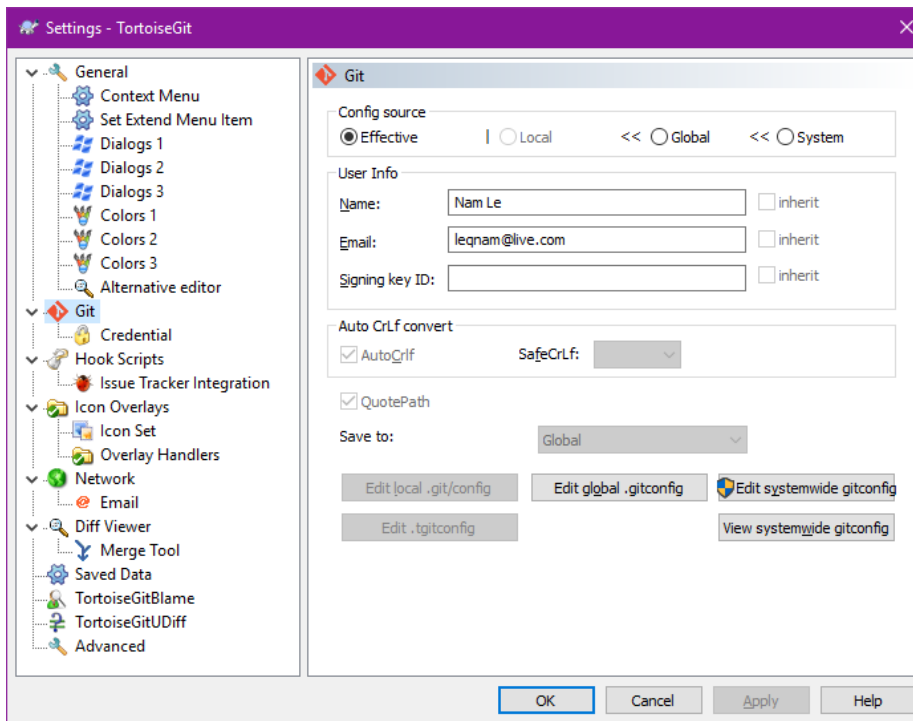
Ngoài ra, có thể dùng TortoiseGit Setting để cấu hình bằng Wizard.



Nhập vào Name và Email

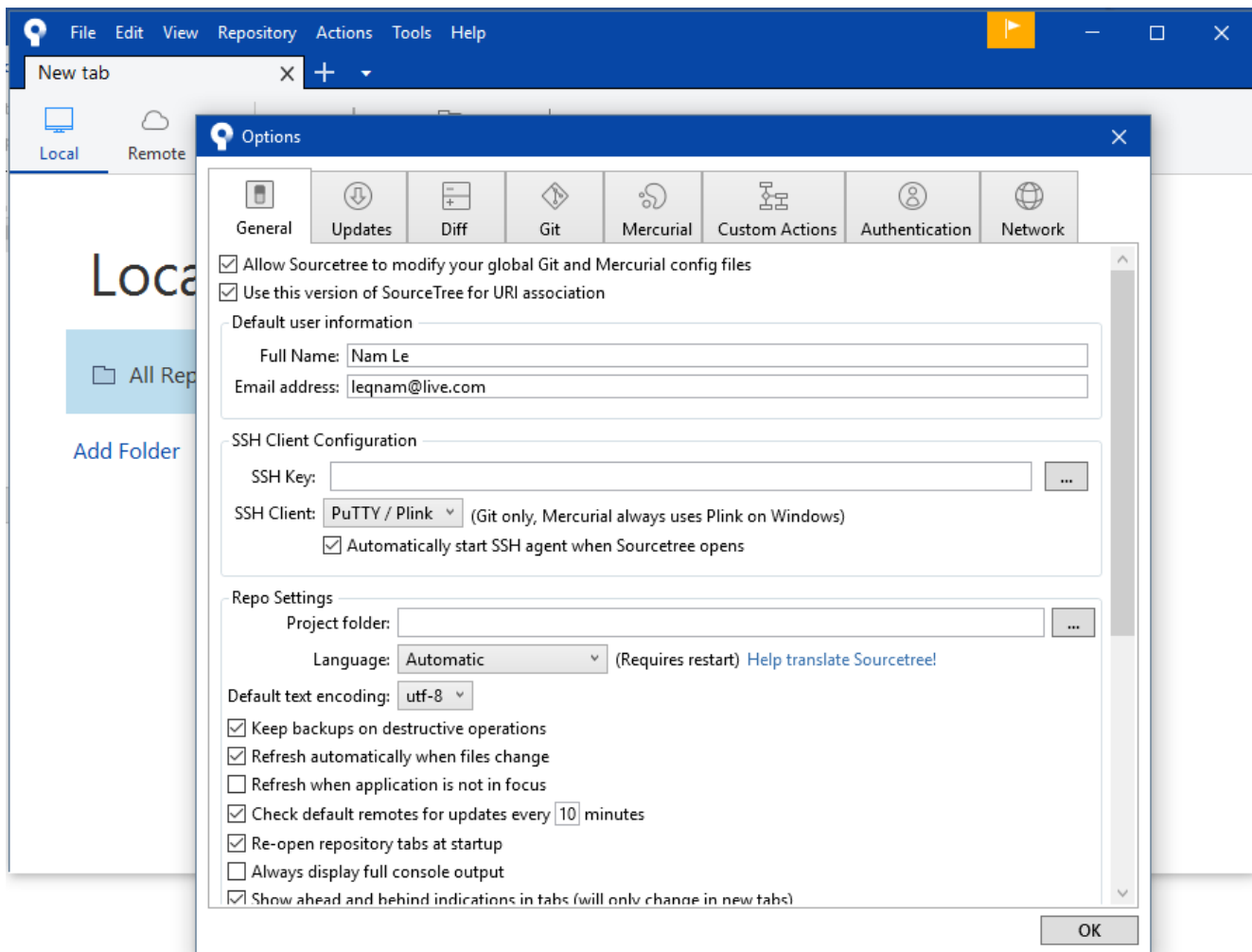
A screenshot of the 'First Start Wizard - TortoiseGit' dialog box. The title bar is purple with the TortoiseGit logo and a close button. The main area has a light gray background. At the top, it says 'Configure user information'. Below this, a message states: 'Git requires that you set up a user name and email address. Both are used as meta data for your commits (not for authentication)'. There are two text input fields: 'Name:' with the value 'Nam Le' and 'Email:' with the value 'leqnam@live.com'. Below the fields, a message says: 'These settings will be stored to your global git configuration (%HOME%/.gitconfig) and will be used for all your git repositories as a default.' There is a checkbox labeled 'Don't store these settings now.' which is currently unchecked. At the bottom, there are four buttons: '< Back', 'Next >', 'Cancel', and 'Help'. The 'Next >' button is highlighted with a blue border.

Các thiết lập này có thể xem lại được ở TortoiseGit Settings



Dùng Sourcetree

Từ Giao diện chính, chọn **Tools** -> **Option**

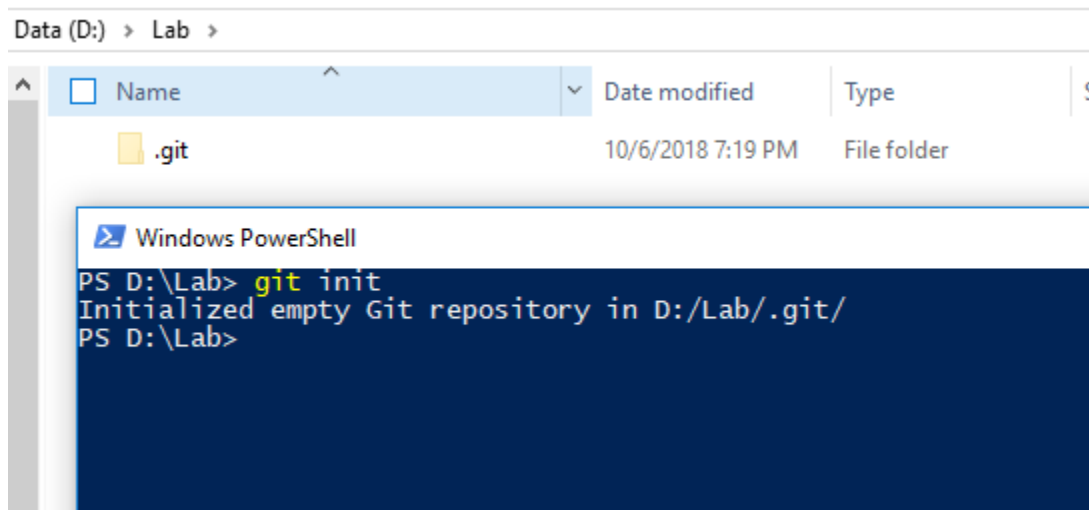


Thao tác với SourceTree đơn giản hơn nhiều.

LÀM VIỆC VỚI GIT

Tạo Local Repository

Sử dụng lệnh `git init` để khởi tạo repository trong thư mục đó. Ví dụ, ta sẽ tạo ra một thư mục mới để chứa code sau này và khởi tạo repository cho nó.



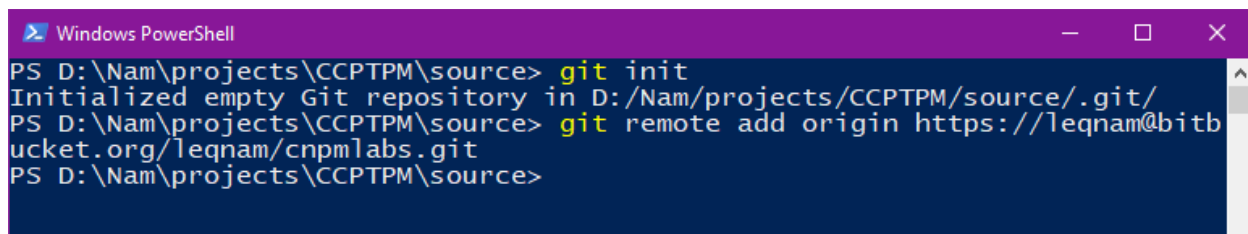
Như vậy là đã tạo được Local Repository. Tuy nhiên chỉ ở trong máy cục bộ, muốn thao tác với Remote Repository thì ta cần phải trỏ Local Repos về Remote Repos.

Kết nối với Remote Repository

Giả sử ta đã có sẵn một repository online tại Bitbucket, có hai tình huống xảy ra:

Thêm mới một (hoặc nhiều) remote repository:

```
git remote add origin https://leqnam@bitbucket.org/leqnam/cnpmlabs.git
```



Thay đổi url của kho lưu trữ từ xa hiện tại

```
git remote set-url origin https://leqnam@bitbucket.org/leqnam/cnpmlabs.git
```

Sau đó push lên remote repos với lệnh

```
git push -u origin master
```

Sao chép (Clone) một Remote Repository

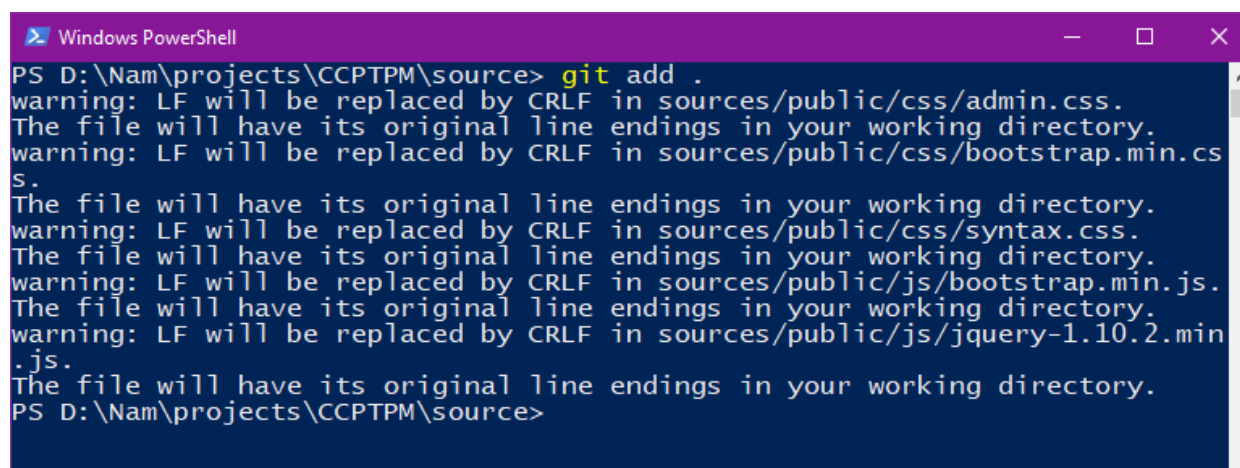
Trường hợp đã có sẵn một Remote Repository, ta không cần tạo Local và Kết nối Remote Repos vào mà ta có thể dùng lệnh git clone để sao chép về:

```
git clone https://leqnam@bitbucket.org/leqnam/cnpmlabs.git
```

Add và Commit

Để có thể làm việc với Git thì ta cần chuyển các tập tin và thư mục thành trạng thái Tracked với lệnh git add. Ta có thể dấu . để gom toàn bộ thay vì từng file riêng lẻ.

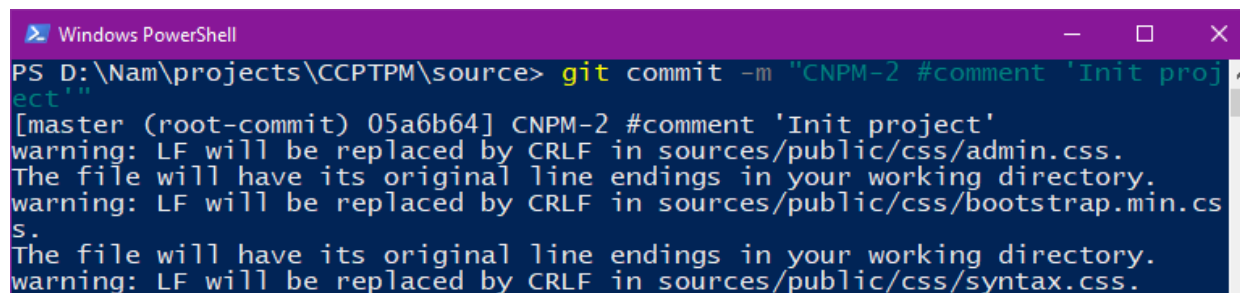
```
git add.
```



```
Windows PowerShell
PS D:\Nam\projects\CCPTPM\source> git add .
warning: LF will be replaced by CRLF in sources/public/css/admin.css.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in sources/public/css/bootstrap.min.css.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in sources/public/css/syntax.css.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in sources/public/js/bootstrap.min.js.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in sources/public/js/jquery-1.10.2.min.js.
The file will have its original line endings in your working directory.
PS D:\Nam\projects\CCPTPM\source>
```

Sau khi add, ta tiến hành commit những thay đổi vào trạng thái của Working Copy, lúc này, những thay đổi của ta chỉ nằm ở Local,

```
git commit -m "Noi dung thong diep"
```



```
Windows PowerShell
PS D:\Nam\projects\CCPTPM\source> git commit -m "CNPM-2 #comment 'Init project'"
[master (root-commit) 05a6b64] CNPM-2 #comment 'Init project'
warning: LF will be replaced by CRLF in sources/public/css/admin.css.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in sources/public/css/bootstrap.min.css.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in sources/public/css/syntax.css.
```

Smart Commit

Ta có thể Commit lên một **JIRA Issue** cụ thể, sử dụng smart commit, ở đây ta có thể tự định nghĩa commit template message, nhưng vì đã liên kết với JIRA và sử dụng **Smart Git commit** nên việc filter theo loại sẽ thực hiện trên JIRA:

```
git commit -m "CNPM-2 #comment 'Init project' "
```

với CNPM là **JIRA Key** lúc ta tạo Project trên Bitbucket, -2 là ID của Issue, **#comment** là tham số cần truyền vào là comment sẽ được liên kết trên **JIRA Issue**

Git push - Đưa thay đổi mã nguồn lên remote repository

Sau khi commit, các thay đổi vẫn đang ở trong Working Copy ở Local Remository, để các thay đổi có mặt trên server, cần phải push lên. Sử dụng lệnh git push:

```
git push --set-upstream origin master
```

```
PS D:\Nam\projects\CCPTPM\source> git push --set-upstream origin master
Counting objects: 120, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (118/118), done.
Writing objects: 45% (54/120)
```

Git Log – Xem lại nhật kí của các commit

Sử dụng lệnh git log để xem lại tất cả các commit

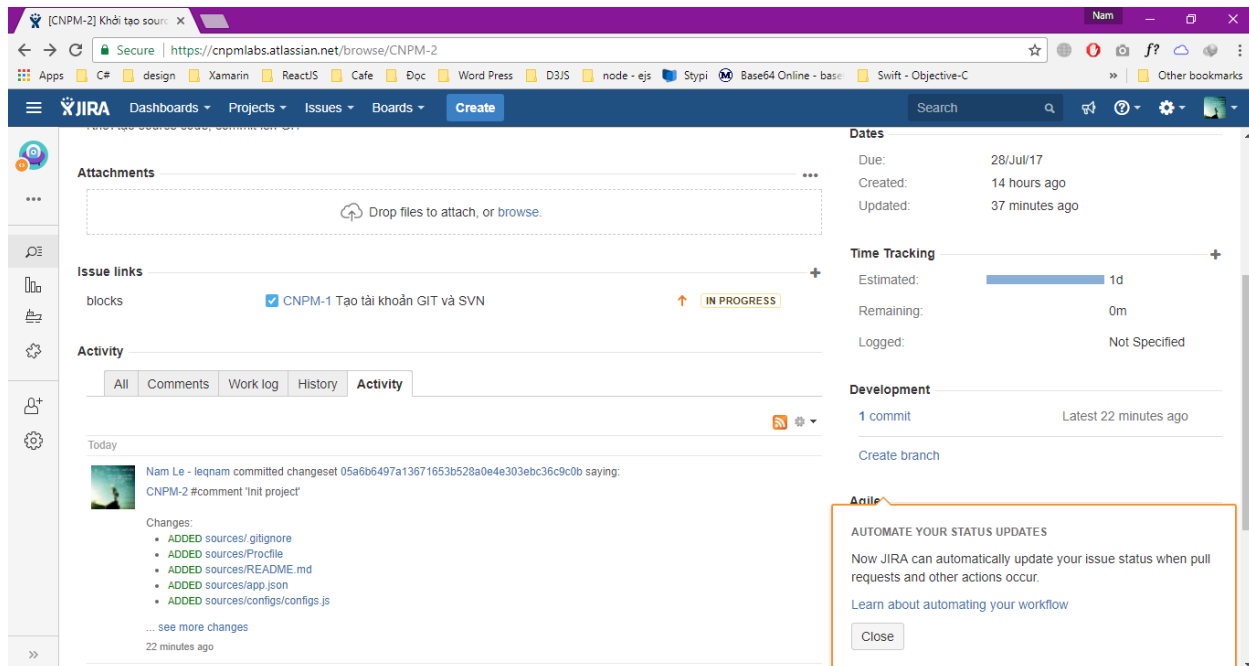
```
git log
```

```
Windows PowerShell
git push --set-upstream origin master

PS D:\Nam\projects\CCPTPM\source> git push --set-upstream origin master
Counting objects: 120, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (118/118), done.
Writing objects: 100% (120/120), 2.01 MiB | 160.00 KiB/s, done.
Total 120 (delta 21), reused 0 (delta 0)
To https://leqnam@bitbucket.org/leqnam/cnpm1abs.git
 * [new branch]      master -> master
Branch master set up to track remote branch master from origin.
PS D:\Nam\projects\CCPTPM\source> git status
On branch master
Your branch is up-to-date with 'origin/master'.
nothing to commit, working directory clean
PS D:\Nam\projects\CCPTPM\source> git log
commit 05a6b6497a13671653b528a0e4e303ebc36c9c0b
Author: Nam Le <leqnam@live.com>
Date:   Thu Jul 27 16:01:30 2017 +0700

    CNPM-2 #comment 'Init project'
PS D:\Nam\projects\CCPTPM\source>
```

Trên JIRA Issue CNPM-2 đã có một commit log:



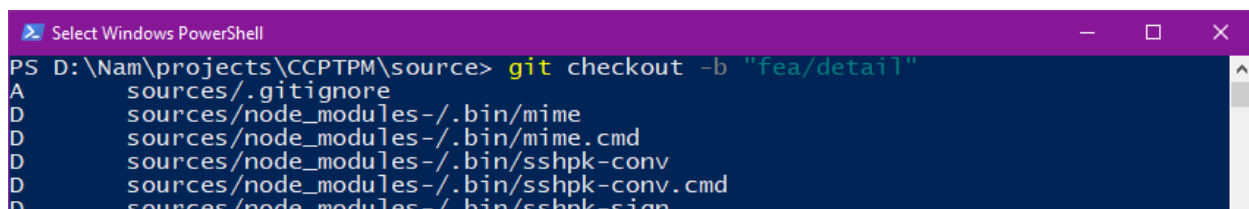
Tạo nhánh

Trong quá trình phát triển phần mềm, cần phải tạo các nhánh để chứa những chức năng cụ thể/ hoặc cho 1 người phát triển sử dụng trong một team, khi cần, chỉ cần “gộp” các nhánh lại để hợp nhất code lại. Việc tạo nhánh để làm việc riêng sẽ giúp tránh được đa số các tình huống bị xung đột (conflict) trong một team có nhiều người cùng làm việc.

Trong trường hợp này có 3 cách tạo nhánh:

1. Tạo trên Remote Repository (ví dụ Bitbucket/ Github,...)
2. Tạo trên công cụ quản lý sự cố (Issue), (ví dụ JIRA/Trello - Một công cụ của Atlassian đi kèm với Bitbucket)
3. Tạo ở Local Repository, sau đó push lên Remote Repository

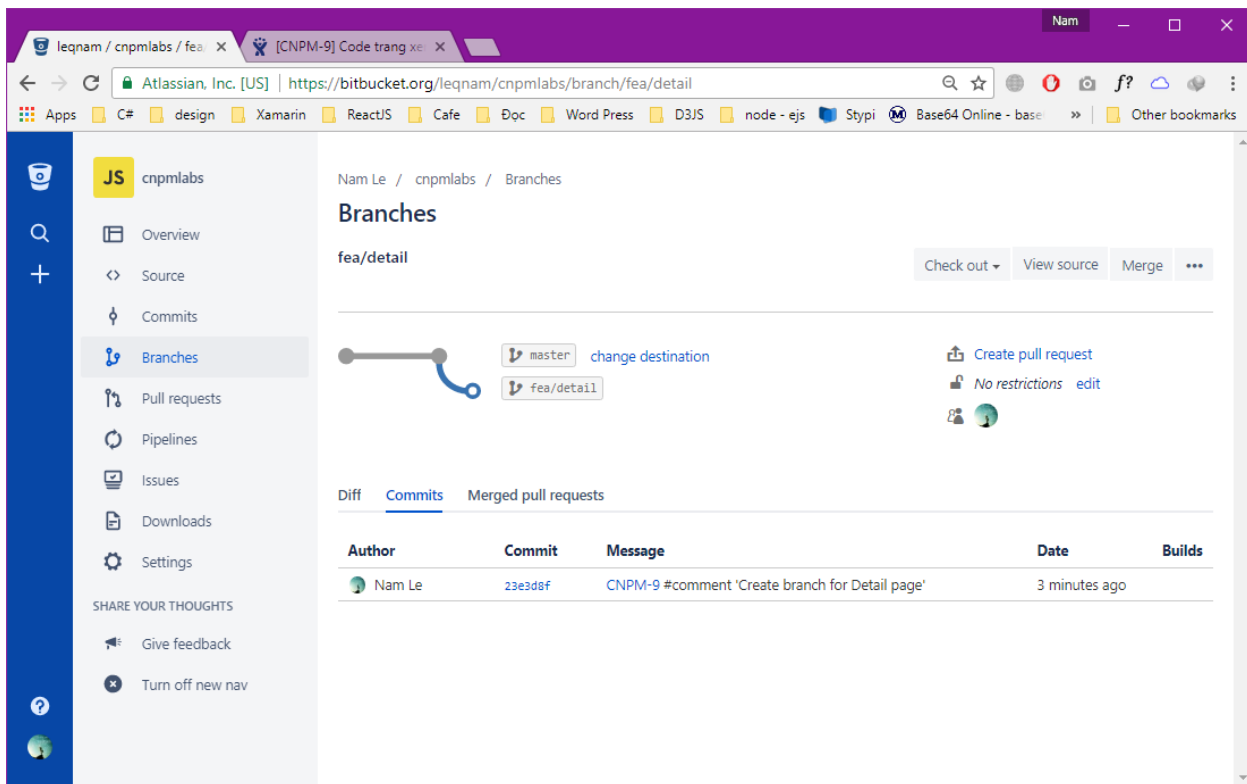
Trong ví dụ sau, Ta sẽ tạo ở local, sử dụng lệnh như hình:



Sau khi tạo nhánh, mặc định working copy đang làm việc sẽ tự chuyển sang nhánh vừa tạo cùng với tất cả các công việc đang làm, ta chỉ cần tiến hành add các file vào hàng đợi, sau đó push code lên nhánh vừa tạo như hình:

```
Windows PowerShell
PS D:\Nam\projects\CCPTPM\source> git push --set-upstream origin fea/detail
Counting objects: 12, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (12/12), done.
Writing objects: 100% (12/12), 4.17 KiB | 0 bytes/s, done.
Total 12 (delta 5), reused 0 (delta 0)
remote:
remote: Create pull request for fea/detail:
remote:   https://bitbucket.org/leqnam/cnpm1abs/pull-requests/new?source=fea/detail&t=1
remote:
To https://leqnam@bitbucket.org/leqnam/cnpm1abs.git
 * [new branch]      fea/detail -> fea/detail
Branch fea/detail set up to track remote branch fea/detail from origin.
PS D:\Nam\projects\CCPTPM\source>
```

Truy cập vào Bitbucket, nhánh đã được tạo ra:



Tạo pull request/ merge nhánh vào nhánh chính (master)

Thông thường có hai cách để “gộp (merge) một nhánh vào nhánh chính (master):

- Dùng lệnh git merge
- Dùng chức năng pull request

Dùng chức năng **pull request** thì sẽ có thể dễ dàng theo dõi thay đổi của source code, từ đó quyết định có merge code từ nhánh phụ về nhánh master hay không, quản lí conflict (xung đột) từ trước khi merge vào master, hạn chế việc người Team leader đảm nhiệm quá nhiều công việc xử lí conflict, người tạo pull request của chính mình cần phải tự giải quyết conflict rồi sau đó tạo pull request lại.

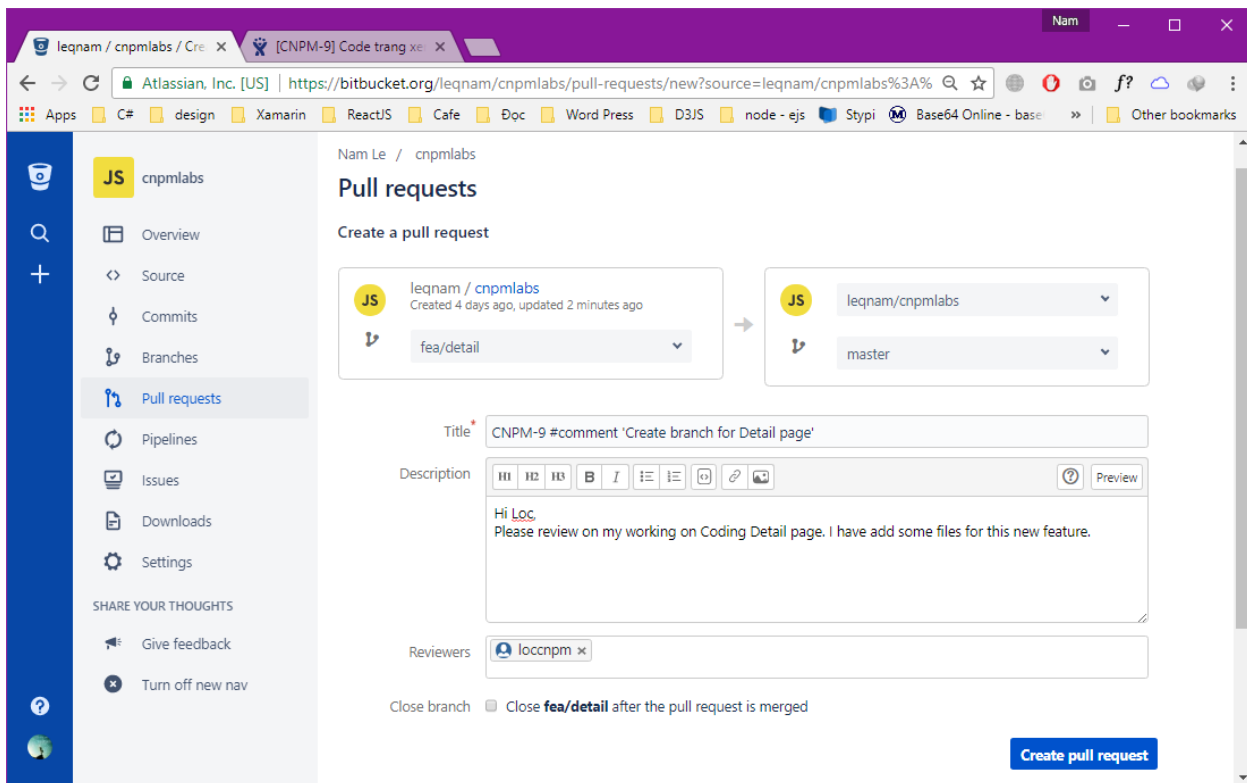
Git merge

Ta có thể gộp một nhánh nào đó vào nhánh mình đang làm việc, vd merge nhánh master vào nhánh fea/detail:

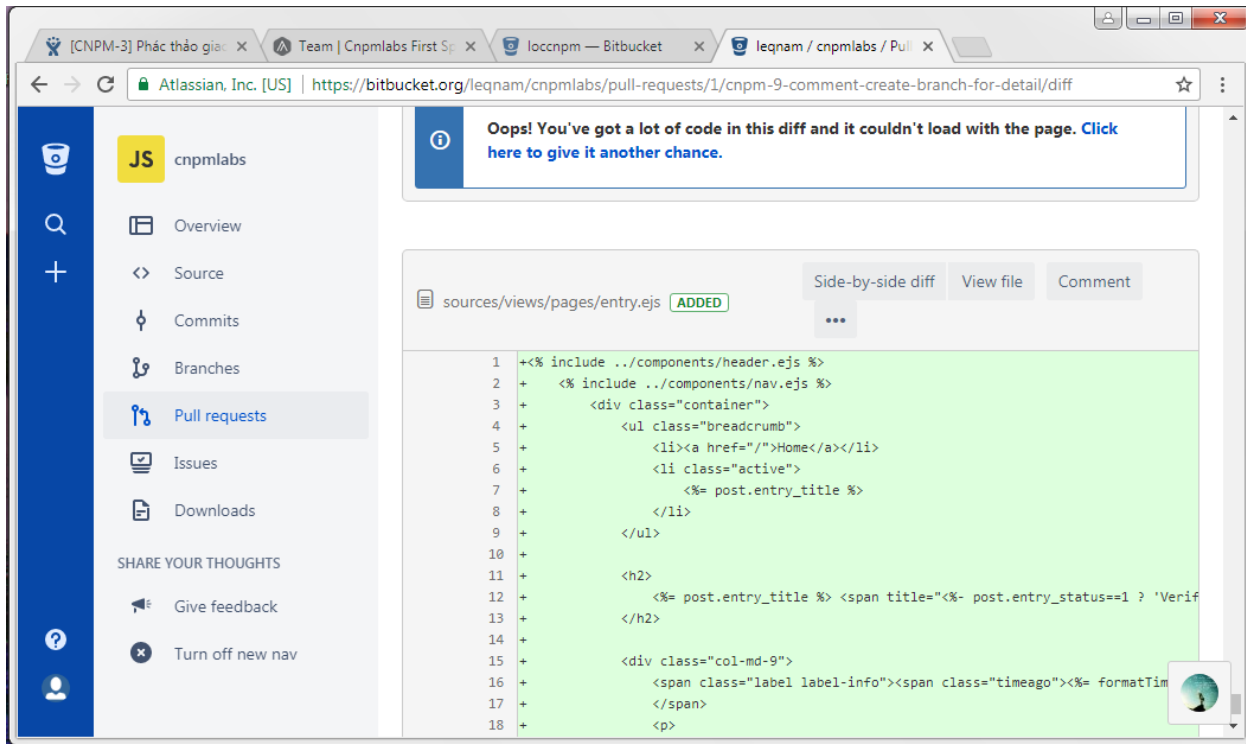
```
git merge master
```

Tạo pull request

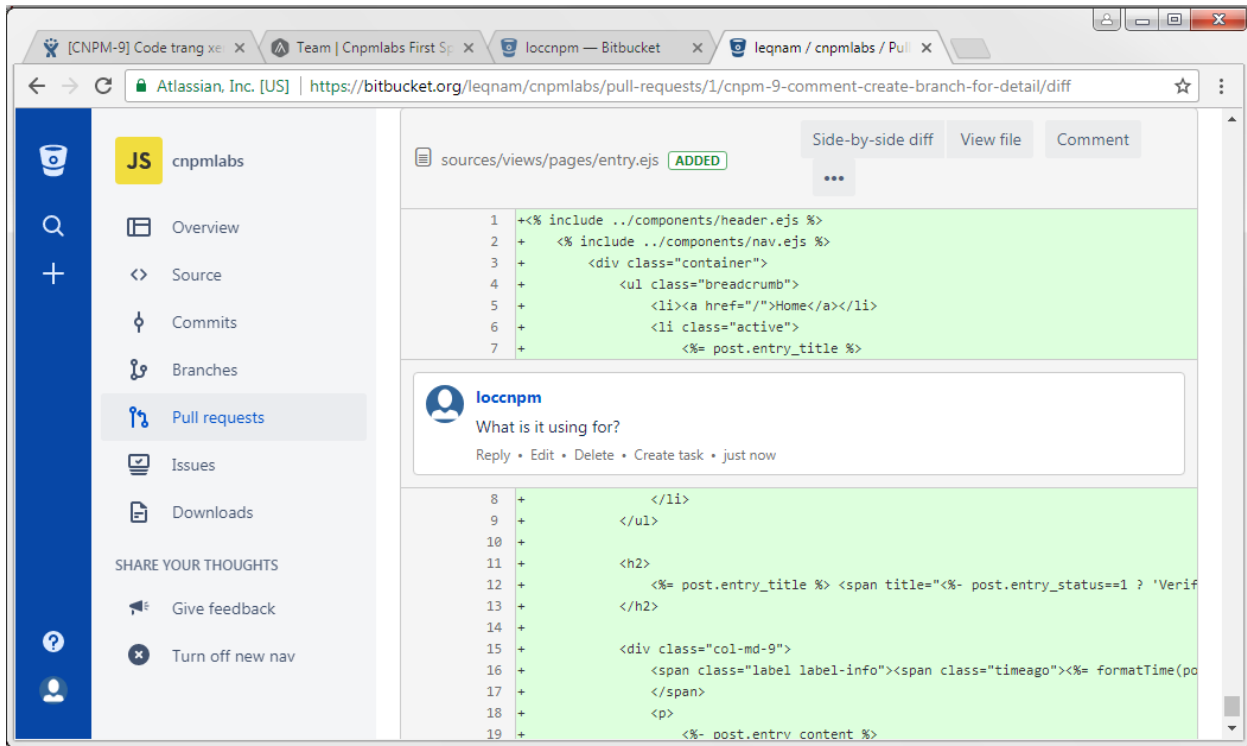
Truy cập vào Bitbucket, chọn Pull request và tiến hành tạo một pull request, chọn nhánh cần merge và nhánh cuối:



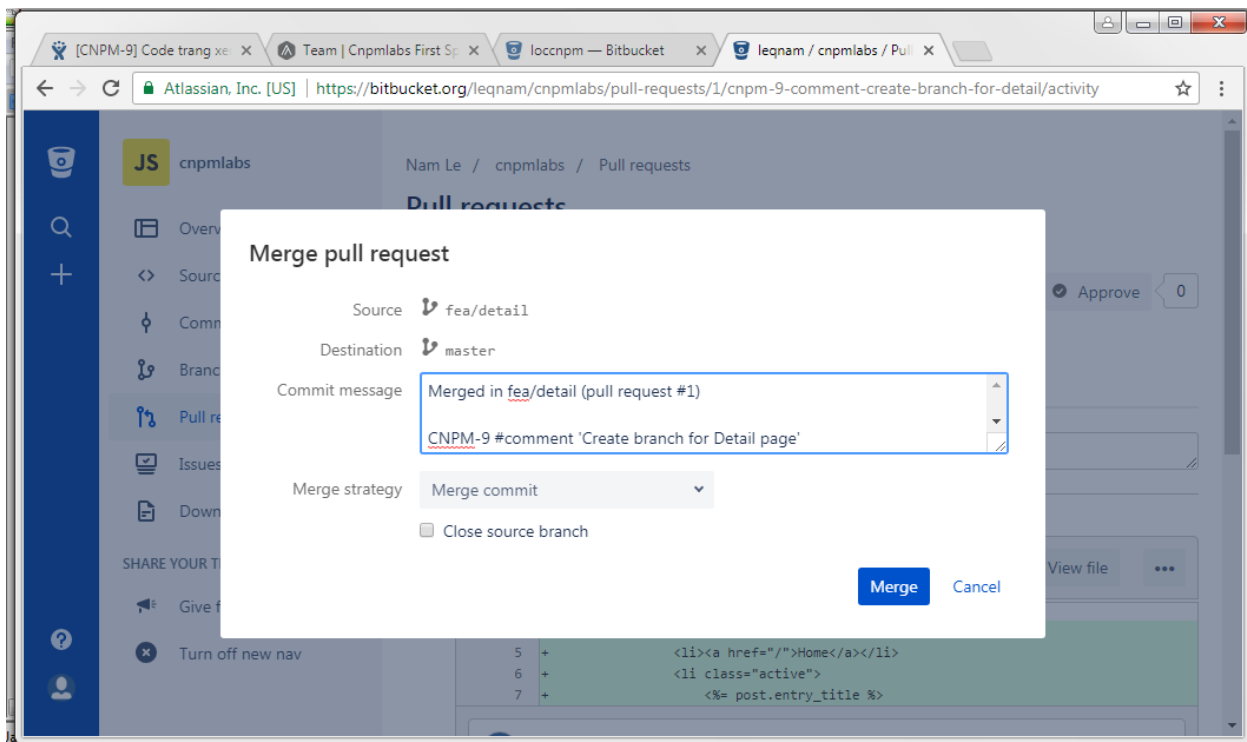
Ở phía người Reviewer khi truy cập vào bitbucket, chọn Pull request sẽ hiện ra các pull request hiện đang có, từ đây sẽ xem nhanh những thay đổi (thêm/xóa/sửa) trong code, file, và quyết định có merge hay không.



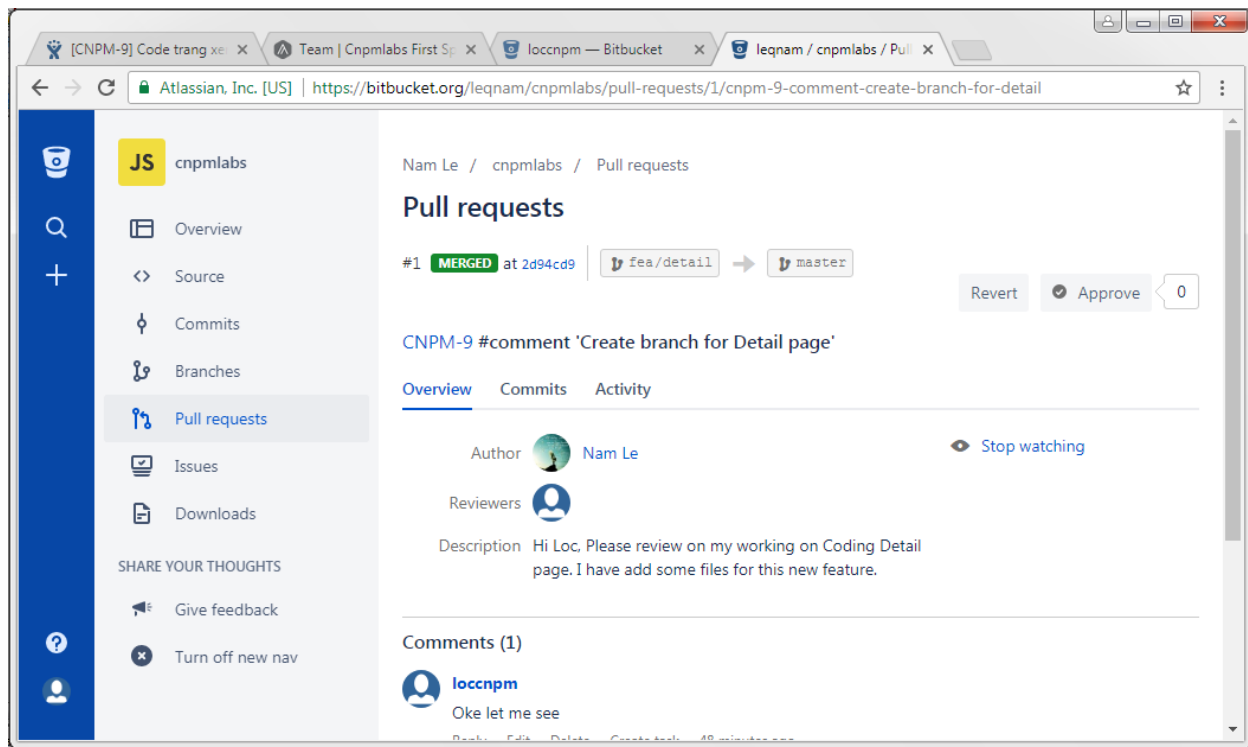
Reviewer cũng có thể để lại một bình luận trên một dòng code nào đó



Sau đó Merge các pull request



Nếu như pull request không có bị conflict thì được merge thành công



Conflict – Xung đột và cách xử lí

Trường hợp 2 user cùng làm việc trên một nhánh, hoặc tạo **pull request**/commit mà pull request/commit đó có reversion cũ hơn nhánh được merge vào (thường là master) thì sẽ xảy ra hiện tượng conflict.

```
Windows PowerShell
PS D:\Nam\projects\CCPTPM\source> git add /
fatal: /: '/' is outside repository
PS D:\Nam\projects\CCPTPM\source> git add .
PS D:\Nam\projects\CCPTPM\source> git commit -m "CNPM-9 #comment 'Remove unuse code'"
[fea/detail 9682c2e] CNPM-9 #comment 'Remove unuse code'
1 file changed, 2 insertions(+), 9 deletions(-)
PS D:\Nam\projects\CCPTPM\source> git push
To https://leqnam@bitbucket.org/leqnam/cnpm labs.git
! [rejected] fea/detail -> fea/detail (non-fast-forward)
error: failed to push some refs to 'https://leqnam@bitbucket.org/leqnam/cnpm labs.git'
hint: Updates were rejected because the tip of your current branch is behind
hint: its remote counterpart. Integrate the remote changes (e.g.
hint: 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
PS D:\Nam\projects\CCPTPM\source>
```

Lúc này phải pull code về rồi mới có thể push code lên remote repository.

Khi này sẽ xảy ra 2 trường hợp:

Tự động Merge thành công

```
Windows PowerShell
[fea/detail 9682c2e] CNPM-9 #comment 'Remove unuse code'
1 file changed, 2 insertions(+), 9 deletions(-)
PS D:\Nam\projects\CCPTPM\source> git push
To https://leqnam@bitbucket.org/leqnam/cnpm1abs.git
! [rejected]        fea/detail -> fea/detail (non-fast-forward)
error: failed to push some refs to 'https://leqnam@bitbucket.org/leqnam/cnpm1abs.git'
hint: Updates were rejected because the tip of your current branch is behind
hint: its remote counterpart. Integrate the remote changes (e.g.
hint: 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
PS D:\Nam\projects\CCPTPM\source> git pull
Auto-merging sources/configs/services.js
Merge made by the 'recursive' strategy.
sources/configs/services.js | 26
1 file changed, 26 deletions(-)
PS D:\Nam\projects\CCPTPM\source> git push
```

Tự động Merge không thành công

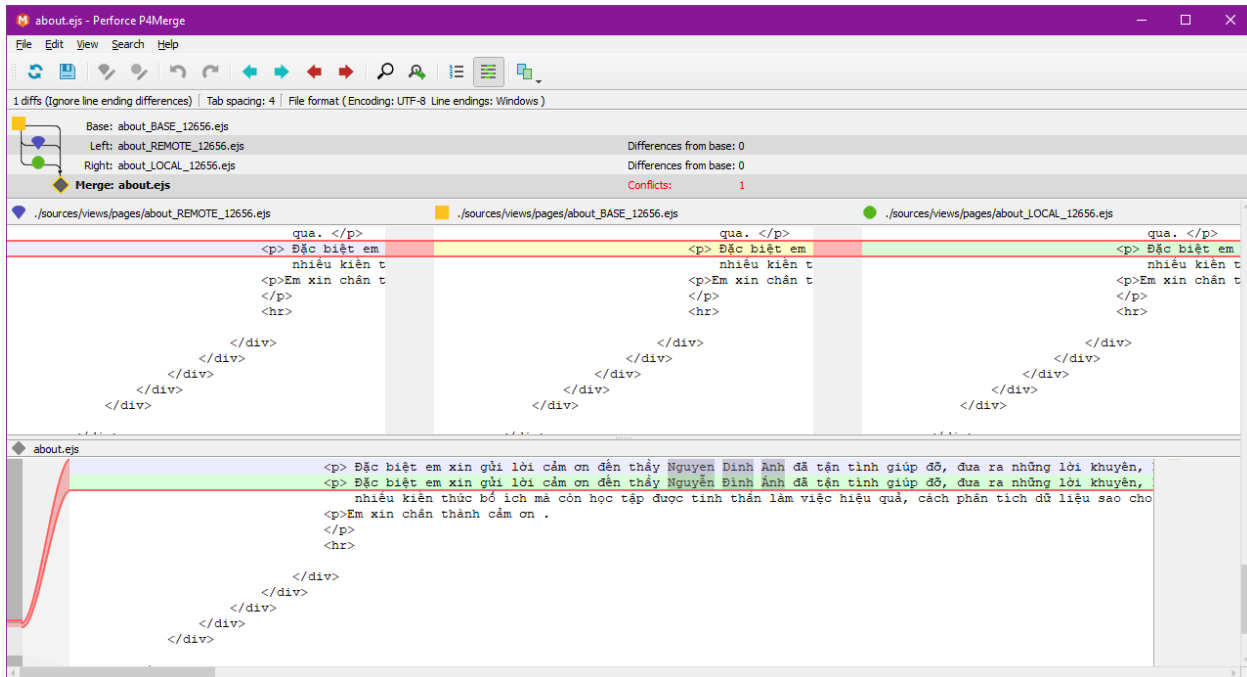
Lúc này, sử dụng lệnh git mergetool để mở chương trình p4merge lên để tiến hành sửa lỗi conflict:

git mergetool

```
PS D:\Nam\projects\CCPTPM\source> git pull
Auto-merging sources/views/pages/about.ejs
CONFLICT (content): Merge conflict in sources/views/pages/about.ejs
Automatic merge failed; fix conflicts and then commit the result.
PS D:\Nam\projects\CCPTPM\source> git merge
error: merge is not possible because you have unmerged files.
hint: Fix them up in the work tree, and then use 'git add/rm <file>'
hint: as appropriate to mark resolution and make a commit.
fatal: Exiting because of an unresolved conflict.
PS D:\Nam\projects\CCPTPM\source>
PS D:\Nam\projects\CCPTPM\source> git mergetool
Merging:
sources/views/pages/about.ejs

Normal merge conflict for 'sources/views/pages/about.ejs':
{local}: modified file
{remote}: modified file
```

Giao diện phần mềm p4merge:



Sau khi sửa các lỗi conflict và tiến hành lưu lại, lúc này có thể push lên remote repository

```
PS D:\Nam\projects\CCPTPM\source> git add .
PS D:\Nam\projects\CCPTPM\source> git commit -m "CNPM-11 #comment 'Change About inf
ormation with Vietnamese' "
[origin 8cf22ee] CNPM-11 #comment 'Change About information with Vietnamese'
PS D:\Nam\projects\CCPTPM\source> git push
fatal: The upstream branch of your current branch does not match
the name of your current branch. To push to the upstream branch
on the remote, use

    git push origin HEAD:master

To push to the branch of the same name on the remote, use

    git push origin origin

To choose either option permanently, see push.default in 'git help config'.
PS D:\Nam\projects\CCPTPM\source> git push origin HEAD:master
Counting objects: 13, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (13/13), done.
Writing objects: 100% (13/13), 1.18 KiB | 0 bytes/s, done.
Total 13 (delta 9), reused 0 (delta 0)
To https://leqnam@bitbucket.org/leqnam/cnpm1abs.git
 772f29c..8cf22ee HEAD -> master
PS D:\Nam\projects\CCPTPM\source>
```

Trường hợp này, ta có thể dùng tính năng có sẵn trong Git-SCM, đó là:

- **Mark as Resolve:** Đánh dấu trạng thái của xung đột là đã được giải quyết
- **Using mine:** Dùng phiên bản thay đổi của chính mình, hành động này sẽ ghi đè các đoạn mã gây conflict của người khác đã có trên server.

- **Using their:** Dùng phiên bản thay đổi của server, hành động này sẽ ghi đè các đoạn mã gây conflict của mình bằng đoạn mã đã có trên server.

Revert commit

Thao tác này được dùng để hủy bỏ một commit đã có trên Remote Repository và quay về commit cụ thể trước đó vì một lí do nào đó.

Ví dụ ta cần reset tất cả commit sau commit có SHA 5e7a720c7 có Description “Changed Validator Service to Constant”

Current Branch		Jump to:	
Graph	Description	Commit	Author
Uncommitted changes		*	*
↳ feat: pod > podS SRI2	↳ origin/feature/...	Me... d6f504c8d	Nam Le
	Renamed Validator SRI const	cca8c4f88	Nam Le
	Merge pull request #2271 in ... from feature/pod...	53a605340	Nam Le
	Merge pull request #2273 in PHOENIX/phoenix from feature/pod...	b08369405	Nam Le
↳ origin/feature/podS/PHOENIX-88488-Display-Detail-Swift	↳ feature	2a44abc1b	Nam Le
	Merge branch 'feature/podS/podS-SRI2' of https://git.ap.manulife.com/...	99143f17f	Nam Le
	Changed Validator Service to Constant	5e7a720c7	Nam Le
	Fixed model for SRI - swiftCode	2af5096a2	Nam Le

Dùng lệnh :

```
git reset --hard 537a720c7 // Hard reset về commit có mã SHA là 537a720c7
```

```
git push origin -f // push lên Remote Repository với tham số -f (là force)
```

```

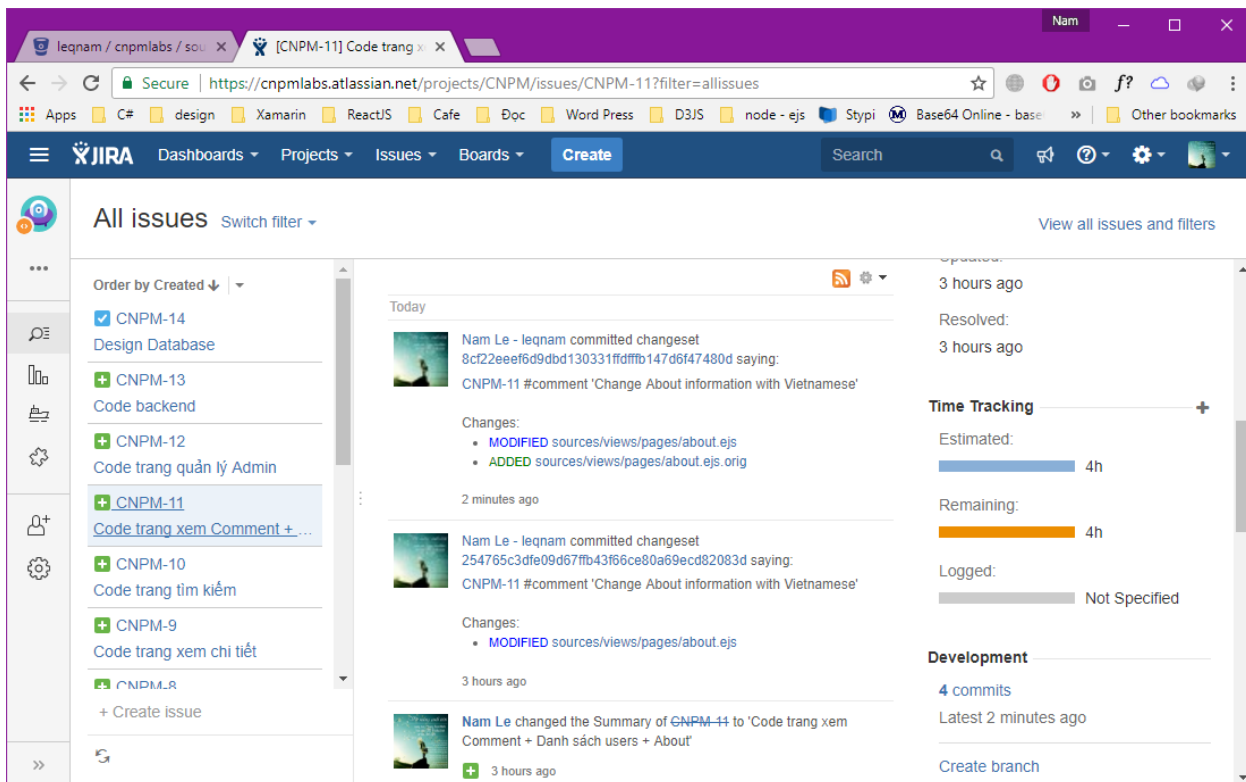
NLe28-Mac:~ nle28$ git reset --hard 5e7a720c7
HEAD is now at 5e7a720c7 Changed Validator Service to Constant
NLe28-Mac:~ nle28$ git push origin -f
Total 0 (delta 0), reused 0 (delta 0)
remote:
remote: View pull request for feature/podS/podS SRI2 => 6046pod:
remote: https://git.ap.manulife.com/projects/PHOENIX/repos/phoenix/pull-requests/2204
remote:
To https://git.ap.manulife.com/scm/phoenix/pdunix.git
+ d6f504c8d...5e7a720c7 feature/podS/podS SRI2 => feature/podS/podS SRI2 (forced update)
NLe28-Mac:~ nle28$

```


Kiểm tra lại bằng lệnh git status:

```
Windows PowerShell
PS D:\Nam\projects\CCPTPM\source> git status
On branch origin
Your branch is up-to-date with 'origin/master'.
nothing to commit, working directory clean
PS D:\Nam\projects\CCPTPM\source>
```

Ta có thể xem trong trang quản lý của Bitbucket/JIRA các commit và nhánh tương ứng với các JIRA Issue:



The screenshot shows the Bitbucket web interface for the repository 'cnpmmlabs'. The left sidebar contains navigation links: Overview, Source, Commits (selected), Branches, Pull requests, Pipelines, Issues, Downloads, Settings, and a section for 'SHARE YOUR THOUGHTS' with links for 'Give feedback' and 'Turn off new nav'. The main content area displays the 'Commits' page for 'Nam Le / cnpmmlabs'. A search bar 'Find commits' is at the top right. Below it, a table lists the commit history with columns: Author, Commit, Message, Date, and Builds. To the left of the table is a commit graph showing the branching structure. The table contains 15 entries, including individual commits and merges.

Author	Commit	Message	Date	Builds
Nam Le	8cf22ee	CNPM-11 #comment 'Change About information with ...	2 minutes ago	
Nam Le	254765c	CNPM-11 #comment 'Change About information with ...	2 hours ago	
locnnp	772f29c	CNPM-11 #comment 'Change About information'	2 hours ago	
Nam Le	25f590e	CNPM-11 #comment 'Code các trang comment'	2 hours ago	
Nam Le	03711d2	CNPM-6 #comment 'Code các trang phụ hoàn tat'	2 hours ago	
Nam Le	8788516	Merged in fea/detail (pull request #3) Fea/detail	2 hours ago	
Nam Le	b550597	Merge branch 'fea/detail' of https://bitbucket.org/leqna...	3 hours ago	
Nam Le	557cc77	CNPM-9 #comment 'Remove commented'	3 hours ago	
locnnp	8e5a08c	CNPM-9 #comment 'Add commented code'	3 hours ago	
Nam Le	c3bdca8	Merge branch 'fea/detail' of https://bitbucket.org/leqna...	3 hours ago	
Nam Le	d9452af	Merged in fea/detail (pull request #2) CNPM-9 #comm...	3 hours ago	
Nam Le	9682c2e	CNPM-9 #comment 'Remove unuse code'	3 hours ago	
locnnp	fb3b45	CNPM-9 #comment 'Remove Commented code'	3 hours ago	
Nam Le	2d94cd9	Merged in fea/detail (pull request #1) CNPM-9 #comm...	4 hours ago	