

Git 基礎介紹

了解 Git 版本控制及 Git GUI 介面使用

什麼是 Git ?

1. Git 是一種版本控制系統，就像是備份的功能，儲存每個階段的檔案狀態

2. 改善我們一直新增新的資料夾
備份資料的缺點



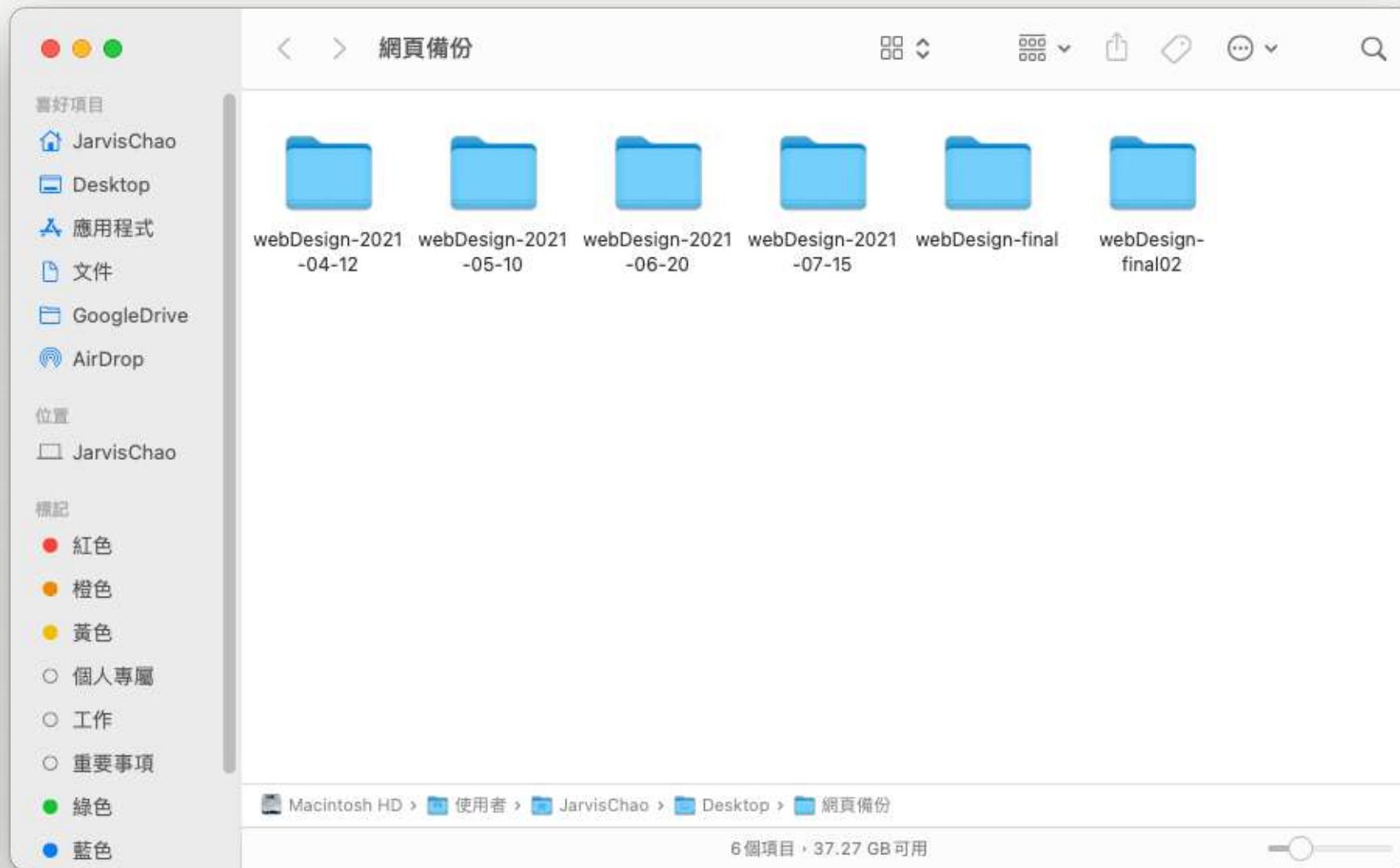
3.

免費、開源、速度快及檔案體積小，只記錄更改過的資料，且可以回到過去

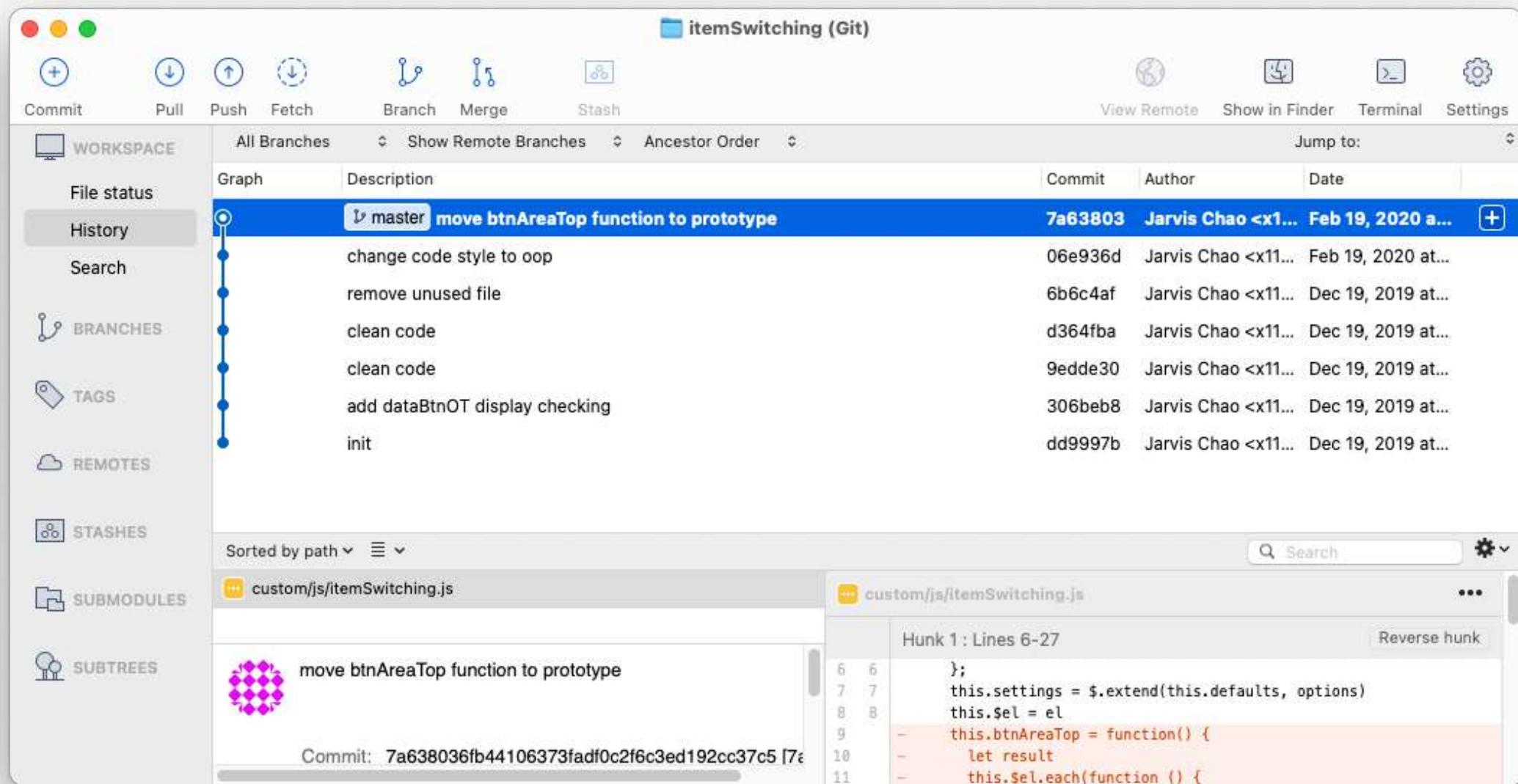
4.

缺點：易學但難專精、指令多且複雜；不過一般使用的指令不太多，懂 20% 的指令就可以應付 80% 的工作了

為什麼要使用 Git?



為什麼要使用 Git?



終端機指令 (Terminal)

建立觀念

VS

圖形介面軟體 (GUI)

方便操作

Mac 常用終端機指令

`pwd`

查詢目前所在資料夾

`mkdir`

建立資料夾

`cd 位置`

進入指定資料夾

`cd ..`

回上一層

`touch`

建立檔案

`ls`

查看檔案清單

`ls -al`

查看檔案清單詳細內容

`rm`

直接刪除檔案，不進垃圾桶

`rm -r`

直接刪除資料夾，不進垃圾桶

`cp 舊檔名 新檔名`

複製檔案成新檔案

`mv 檔名 新位置`

搬移檔案至新位置

`mv 檔名 新檔名`

重新命名檔案至新檔名

`clear`

清除畫面內容

Windows 常用終端機指令

cd

查詢目前所在資料夾

mkdir

建立資料夾

cd 位置

進入指定資料夾

cd ..

回上一層

type nul > 檔名

建立檔案

dir

查看檔案清單詳細內容

dir /w

查看檔案清單詳細內容

del

直接刪除檔案，不進垃圾桶

rmdir /s

直接刪除資料夾，不進垃圾桶

copy 舊檔名 新檔名

複製檔案成新檔案

move 檔名 新位置

搬移檔案至新位置

move 檔名 新檔名

重新命名檔案至新檔名

cls

清除畫面內容

Git 基本設置及指令

初始化 Git 及基本指令

基本設置

\$ git config --list

檢視目前設定

```
$ git config --list
```

\$ mkdir 資料夾名

建立資料夾

```
$ mkdir git-test
```

\$ git config --global user.name "設定值"

設定 username 及 email

```
$ git config --global user.name "Jarvis"  
$ git config --global user.email "xxx@gmail.com"
```

\$ cd 資料夾名

進入資料夾

```
$ cd git-test
```

基本指令

\$ git init

初始化 git 並在指定資料夾建立 .git 檔案

```
$ git init
```

\$ git add

將檔案狀態從**工作目錄**移至**暫存區**

```
$ git add index.html  
$ git add --all // 新增所有修改過的檔案
```

\$ git status

查看 git 工作目錄狀態

```
$ git status
```

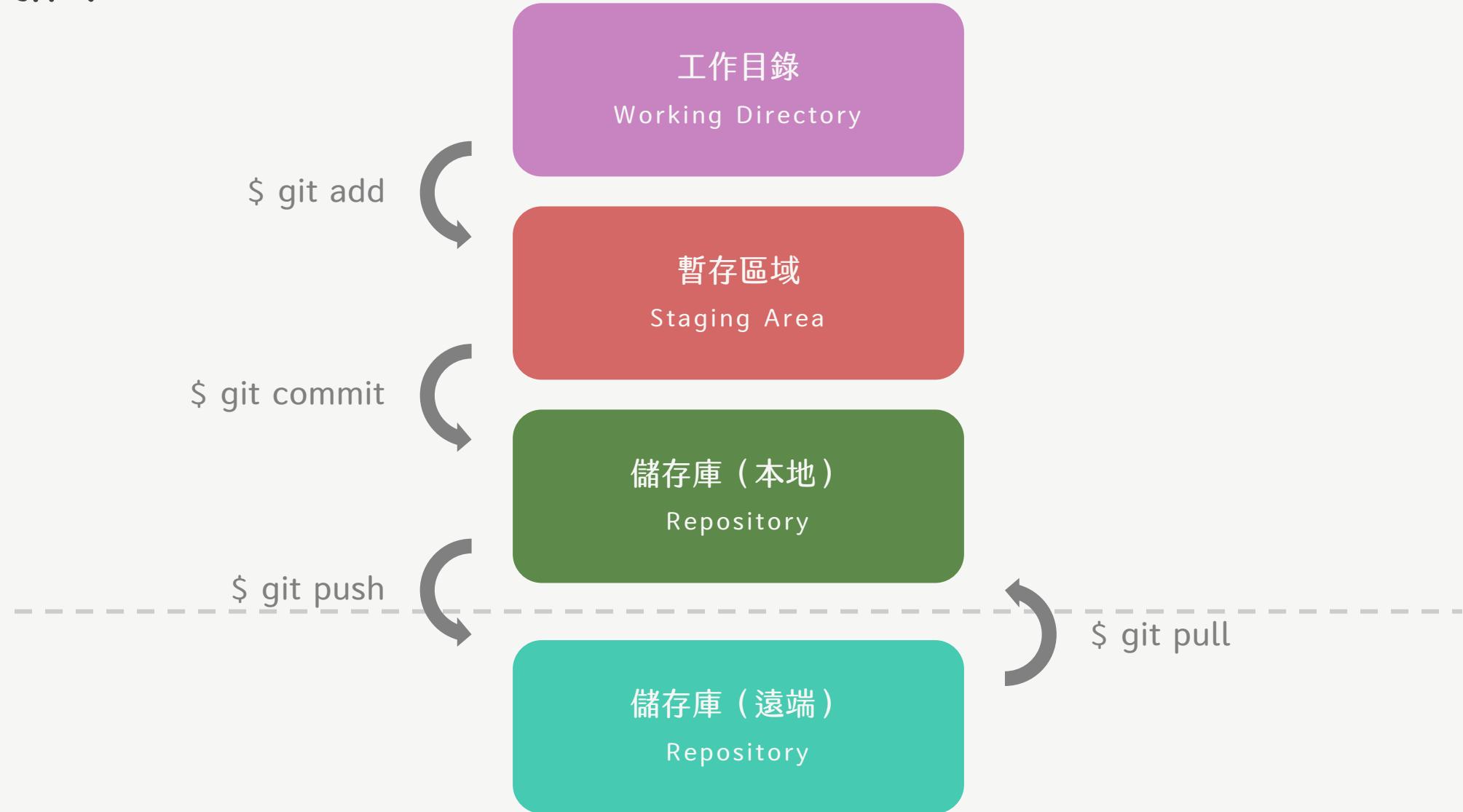
\$ git commit -m "說明文字"

將檔案狀態從**暫存區**移至**本地儲存庫**；說明文字需簡單扼要

```
$ git commit -m "first commit"
```

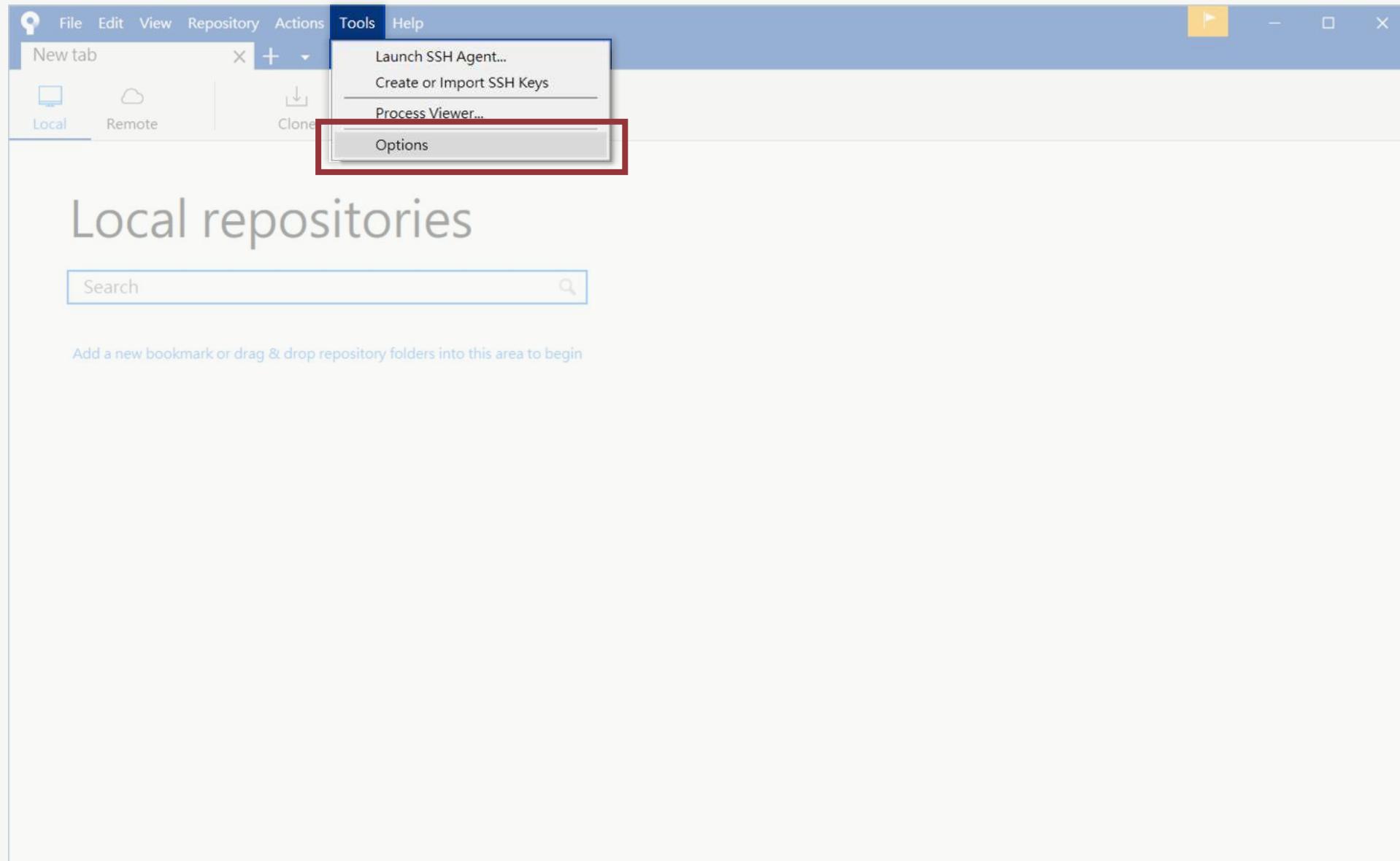
為什麼要兩段式，又是 add 又是 commit？ 就像是囤貨一樣，先將做好的檔案放置倉庫外，等累積一定數量再一次搬進倉庫

基本指令



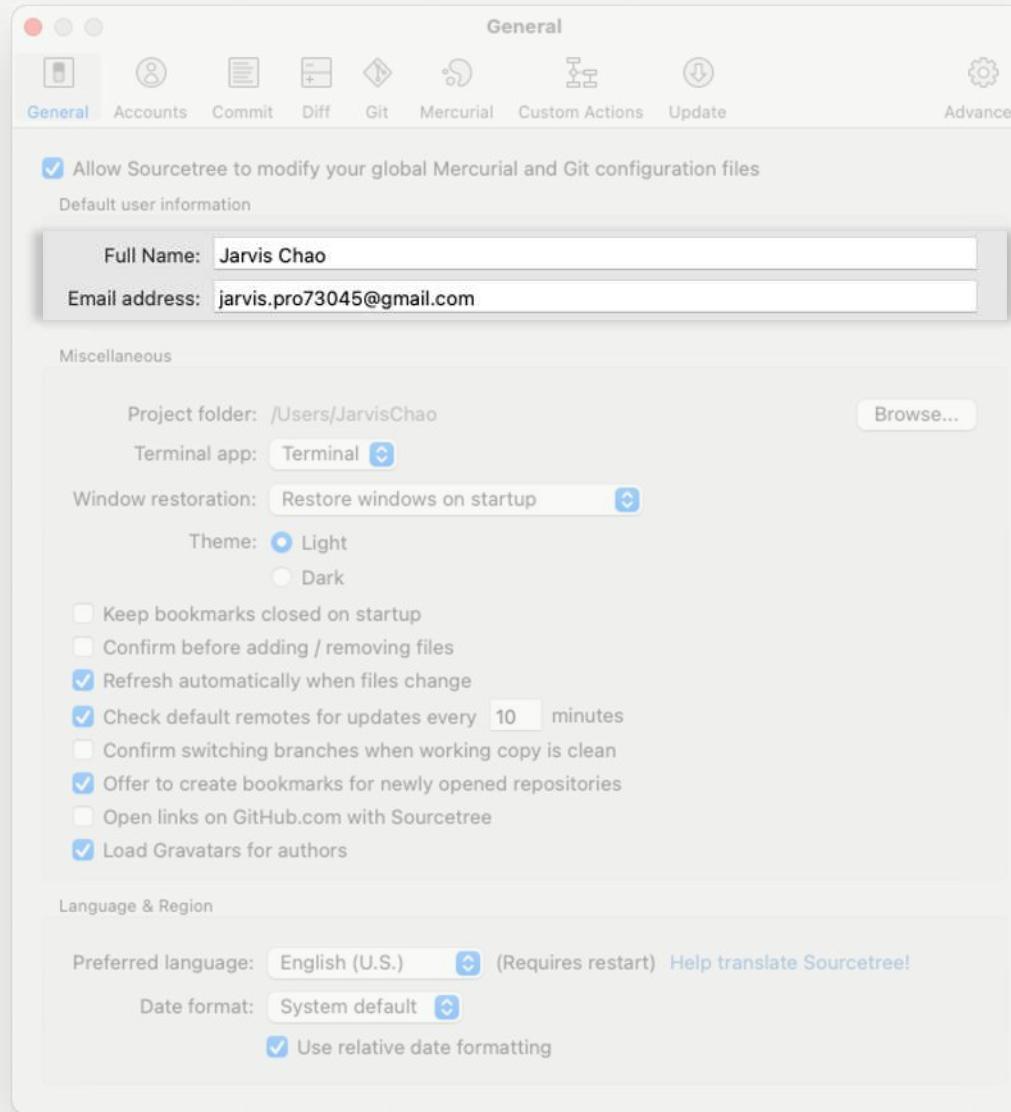
commit 的說明文字 "bug fixed" 與 "update" 不是好訊息

Sourcetree (連結 Github 帳戶)

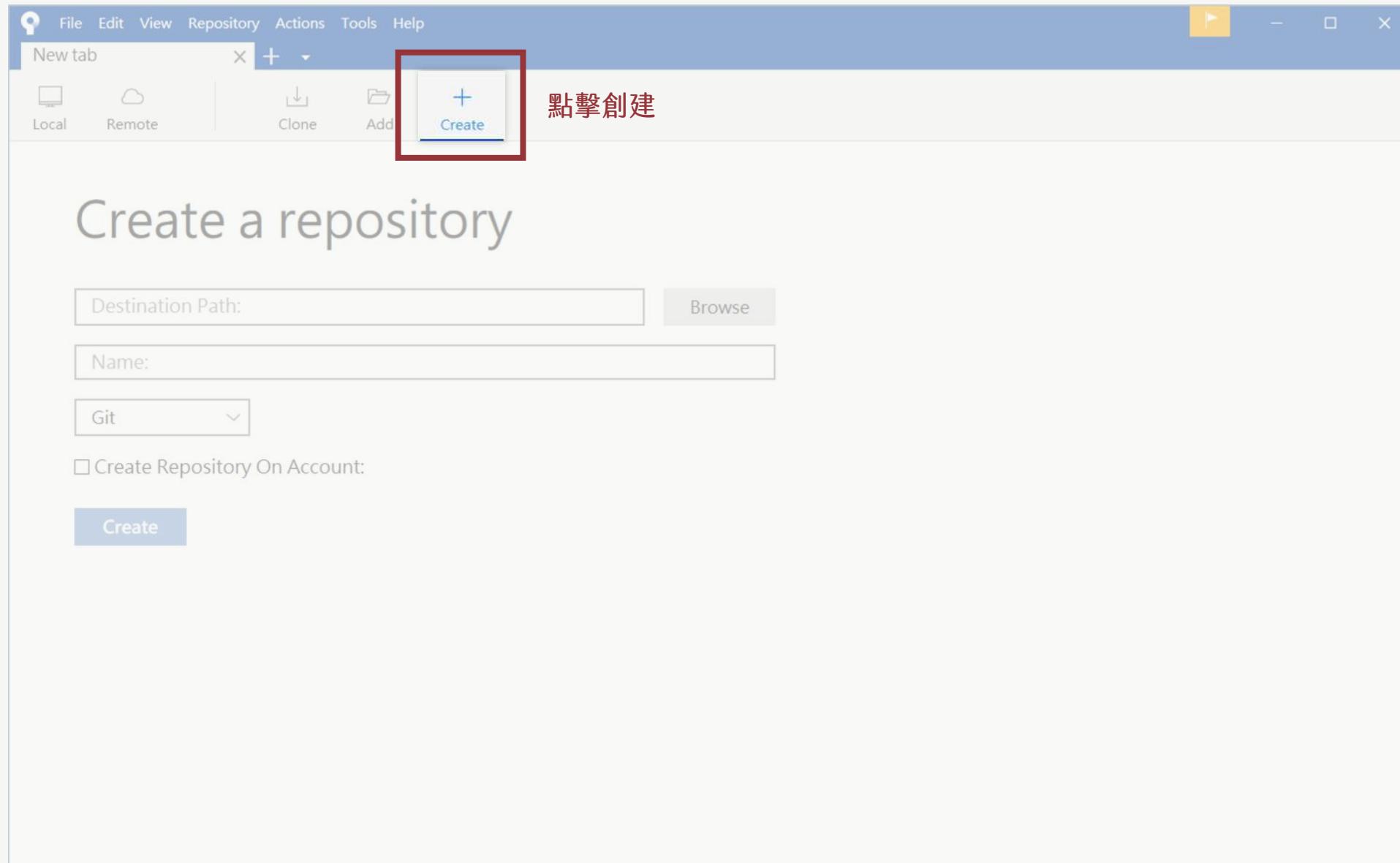


Sourcetree (config)

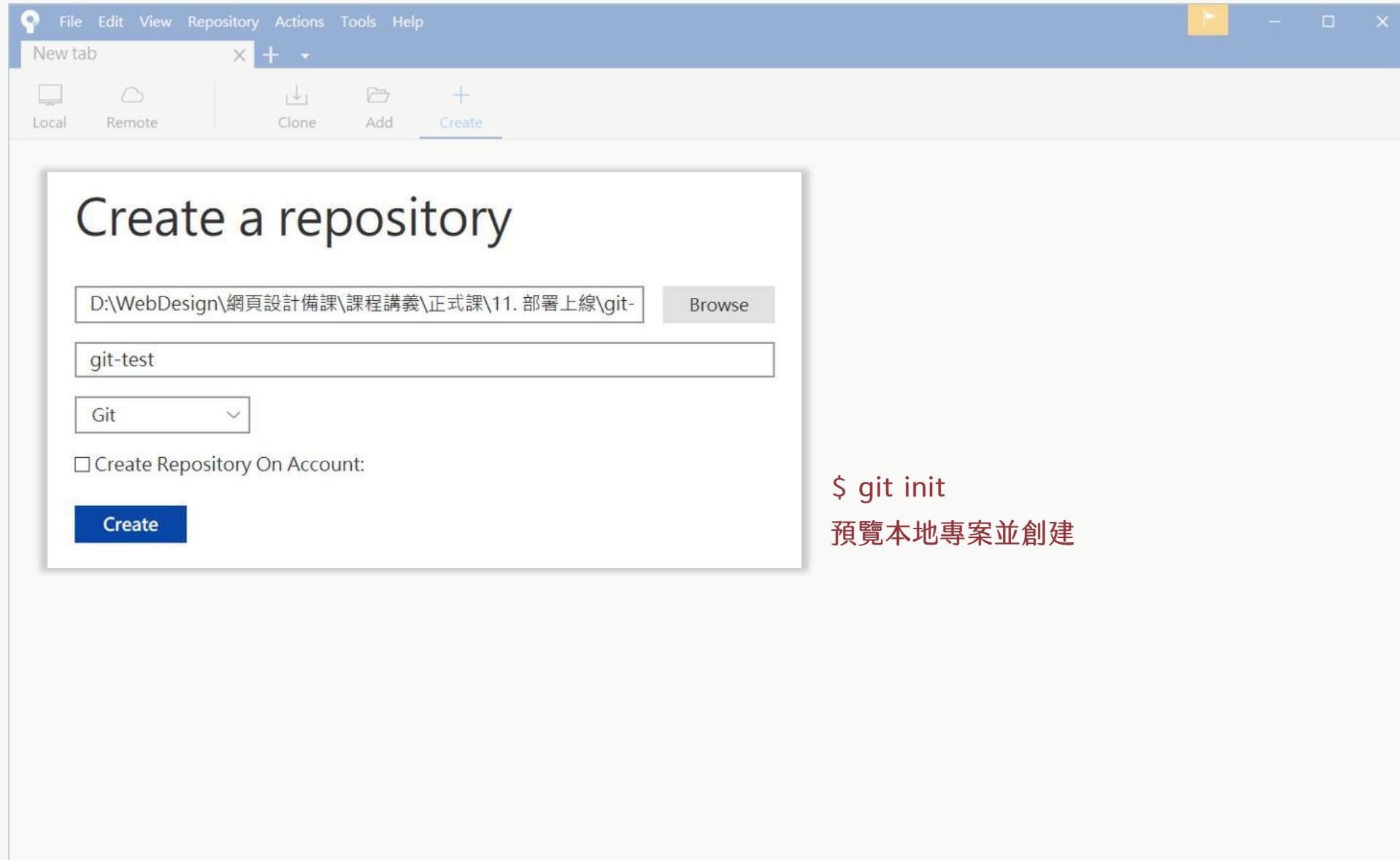
```
$ git config --global user.name  
$ git config --global user.email
```



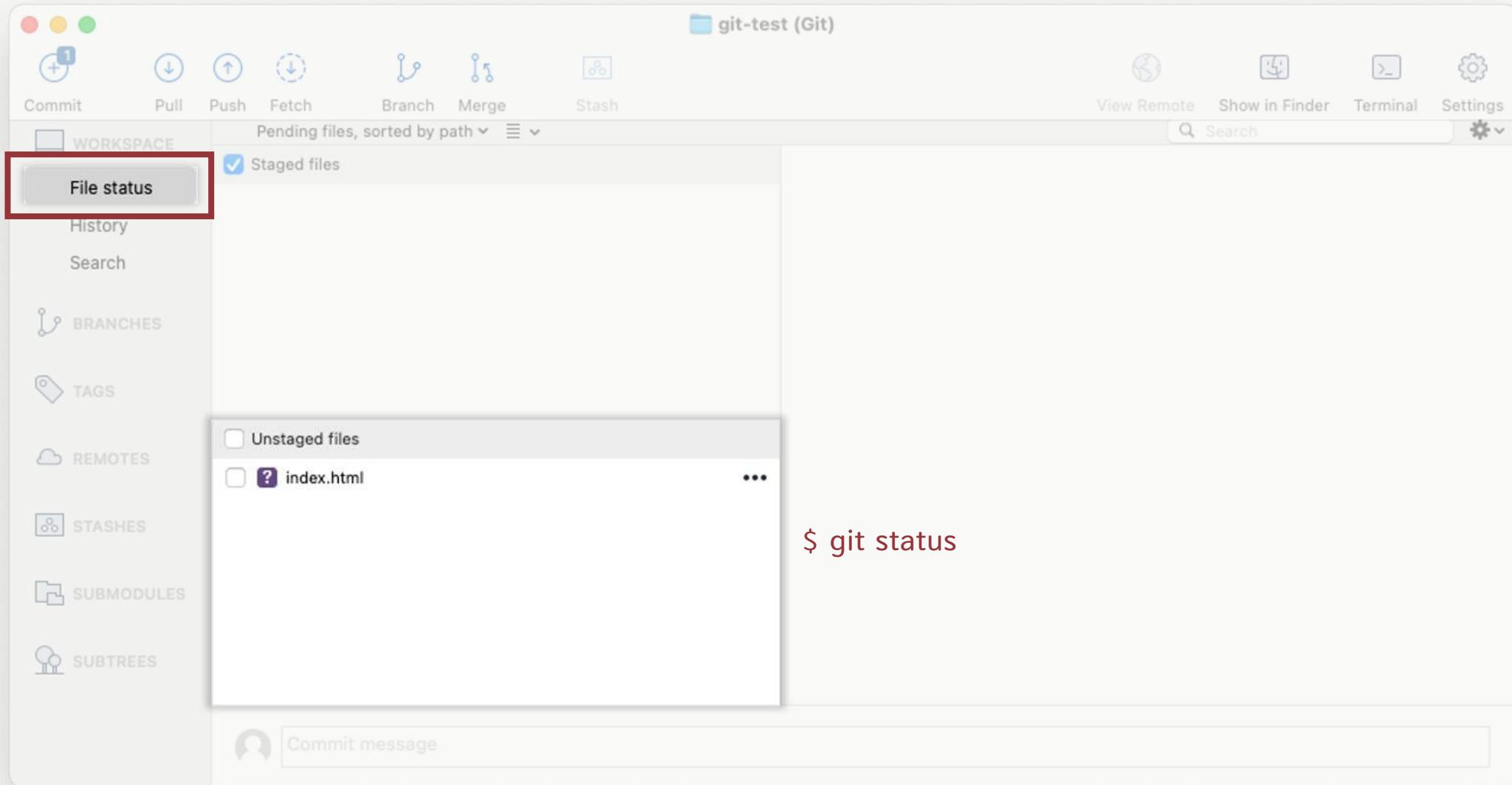
Sourcetree (創建專案)



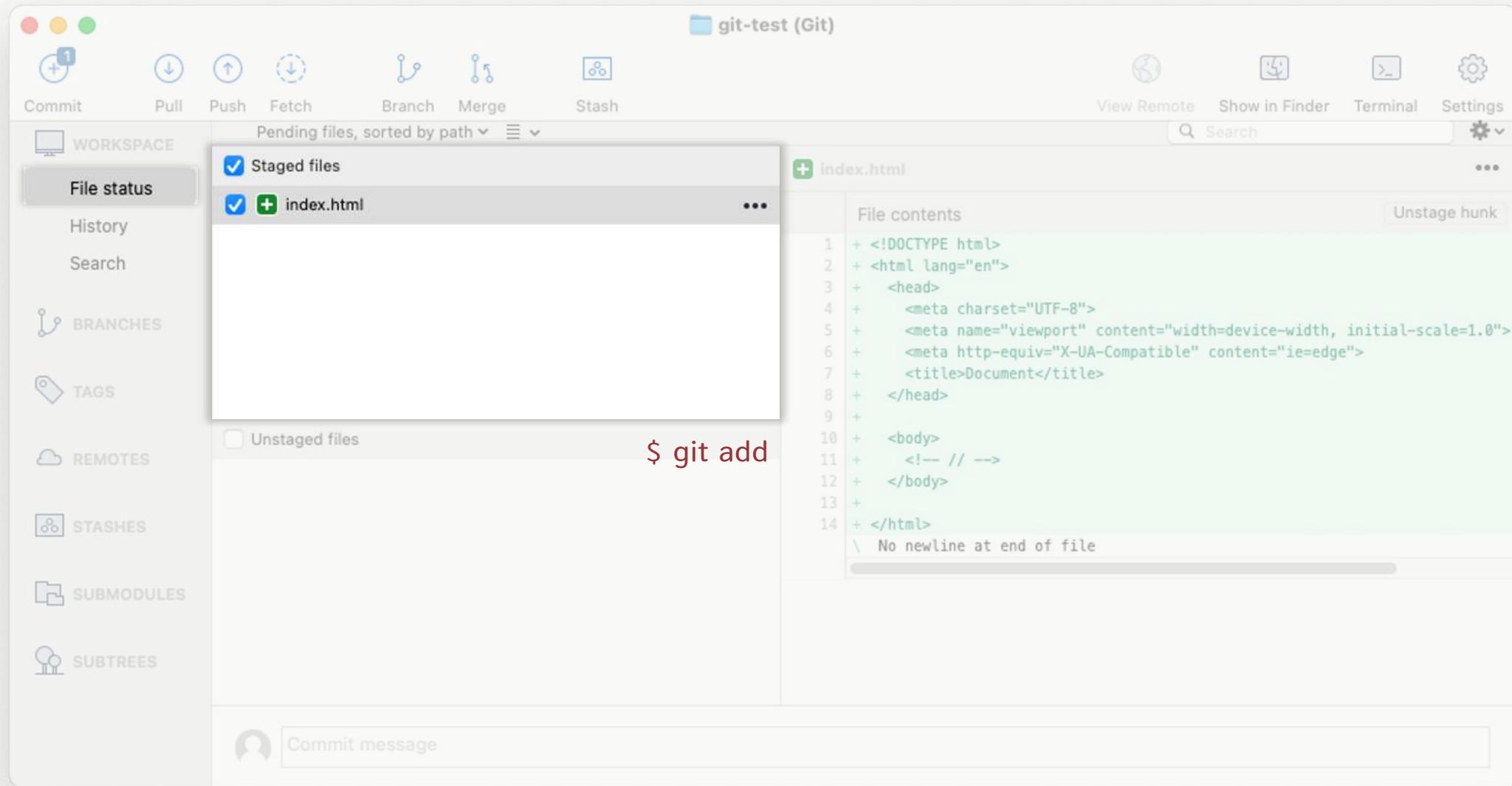
Sourcetree (新建專案)



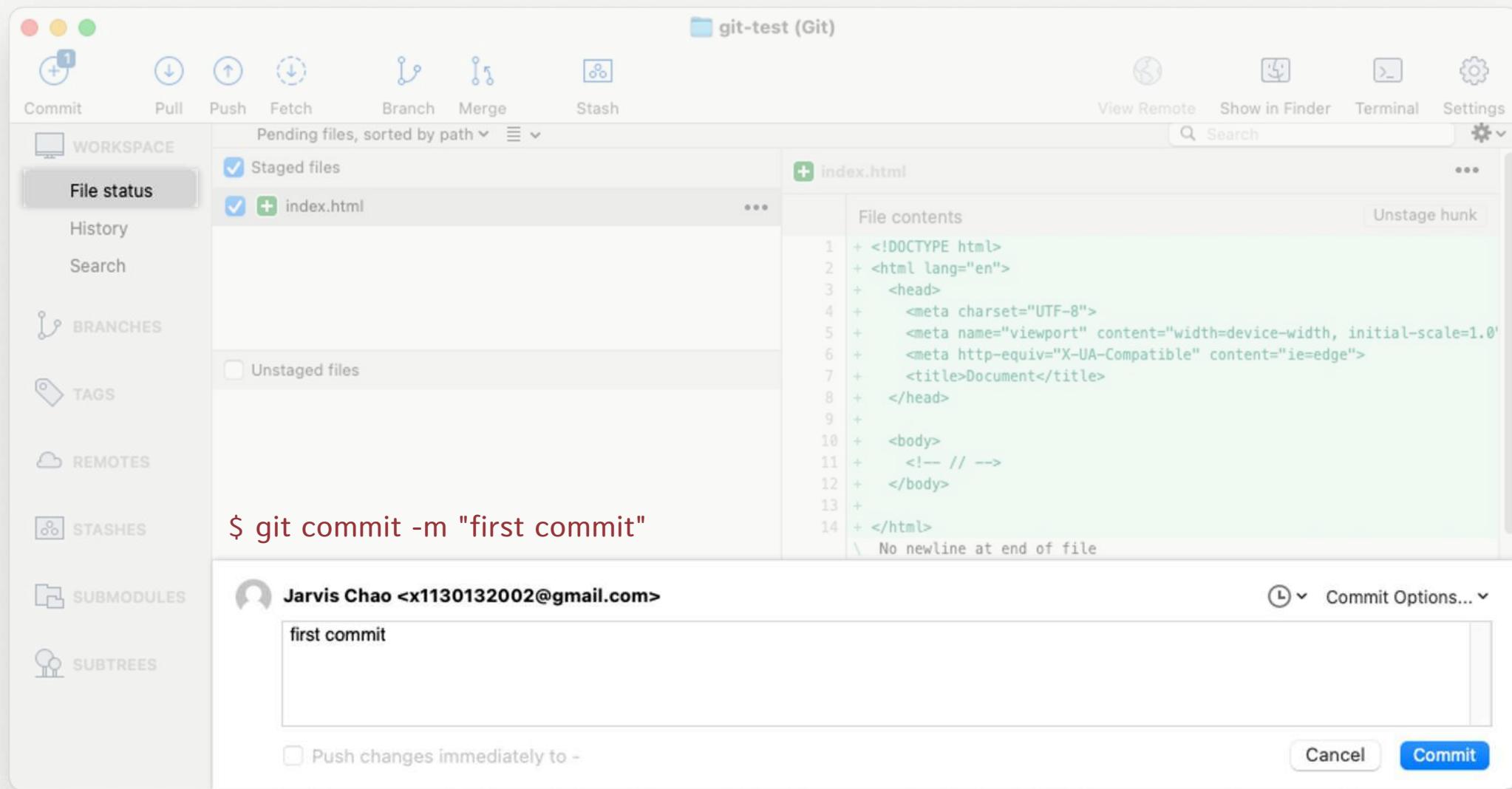
Sourcetree (status)



Sourcetree (add)



Sourcetree (commit)



Git 其他常用指令

其他 Git 常用指令

常用指令

\$ git log

檢視 git 紀錄

```
$ git log  
$ git log --oneline // 簡潔模式
```

\$ git checkout

還原檔案被刪除檔案

```
$ git checkout index.html  
$ git checkout . // 救回全部被刪除的檔案
```

\$ git blame

檢視修改者

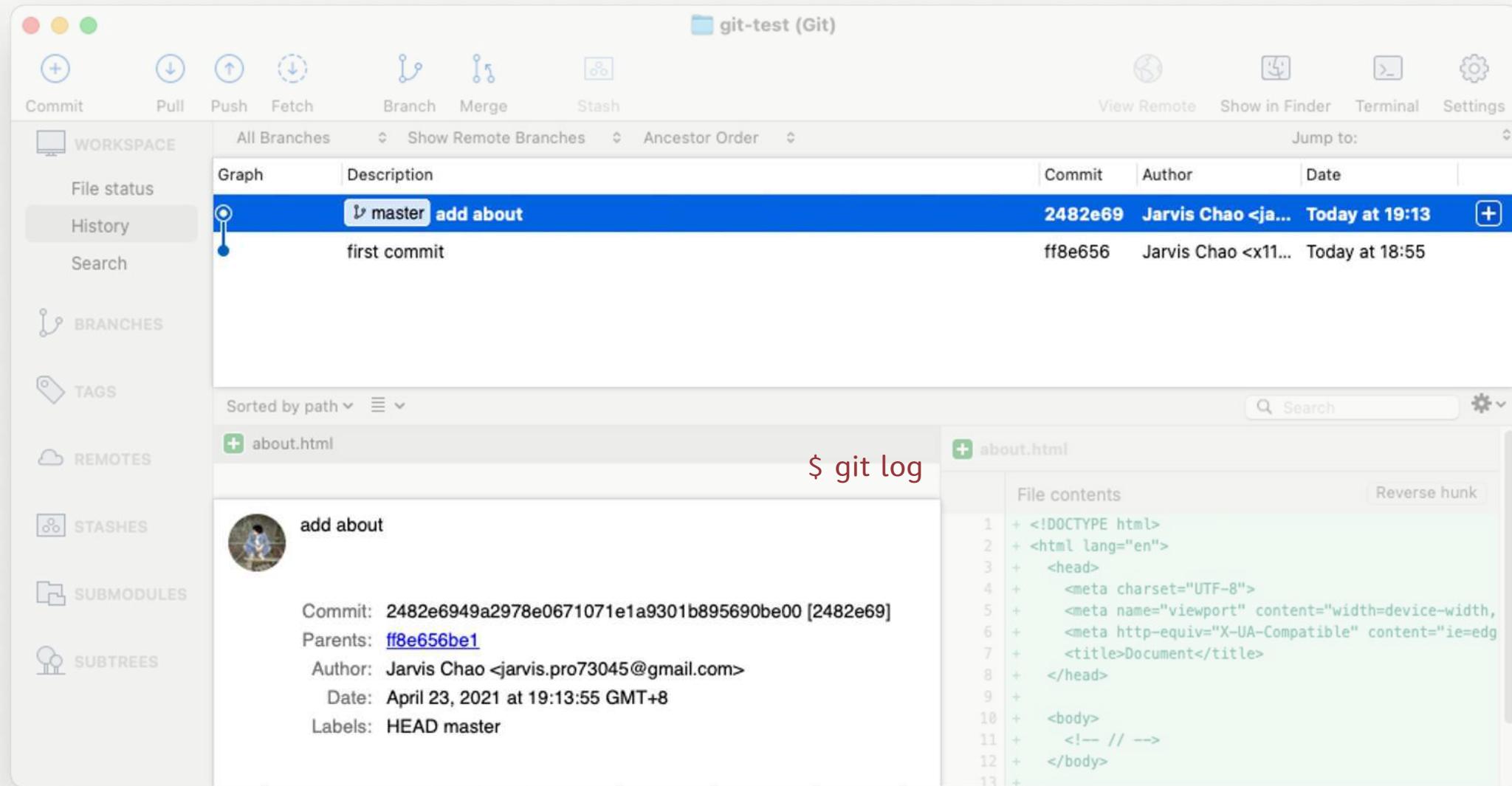
```
$ git blame index.html
```

\$ git stash

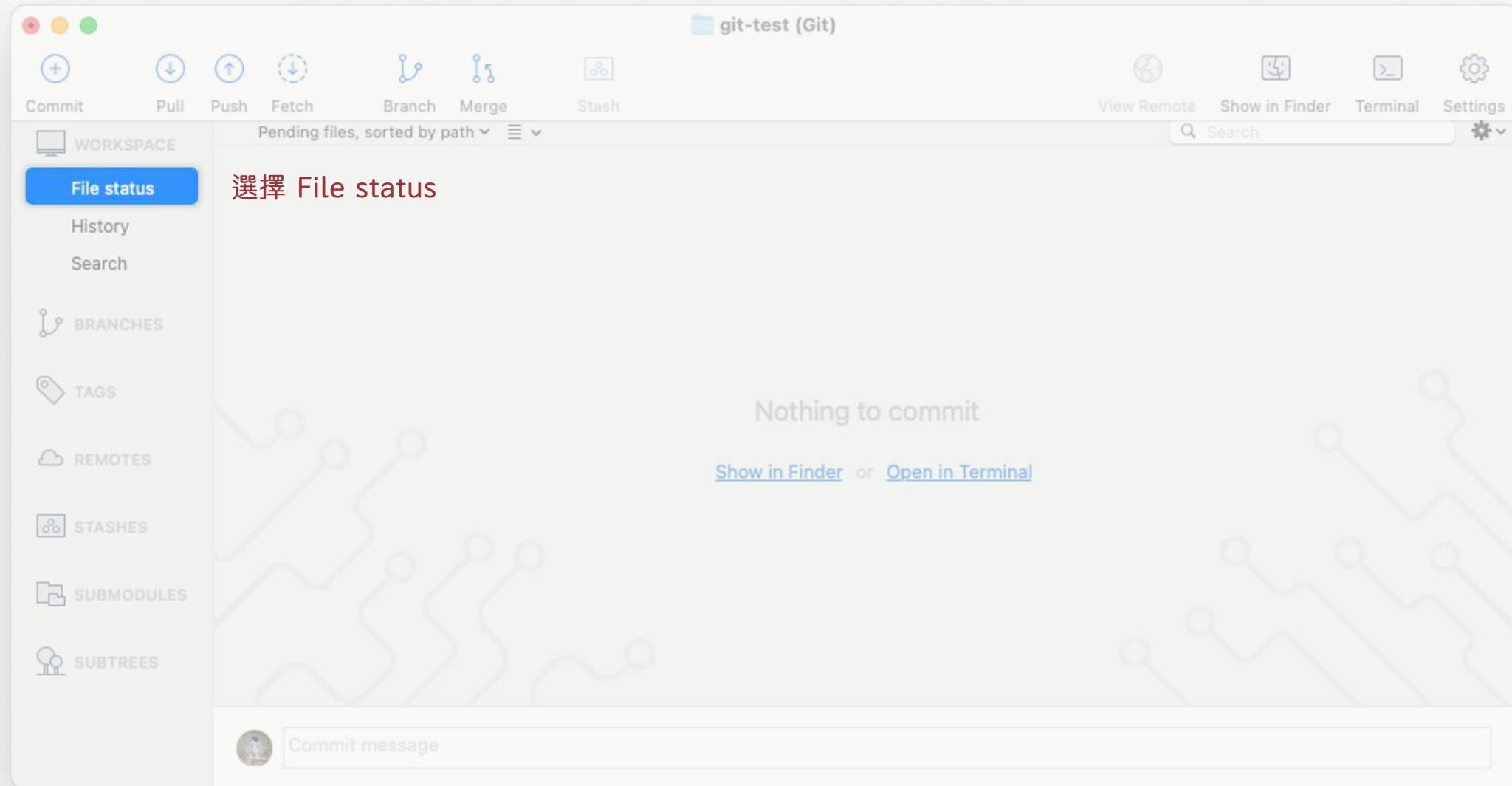
將編輯中的資料緩存起來

```
$ git stash  
$ git stash list // 查看 stash 清單  
$ git stash pop // 將資料拿出並刪除  
$ git stash apply // 將資料拿出並保留住
```

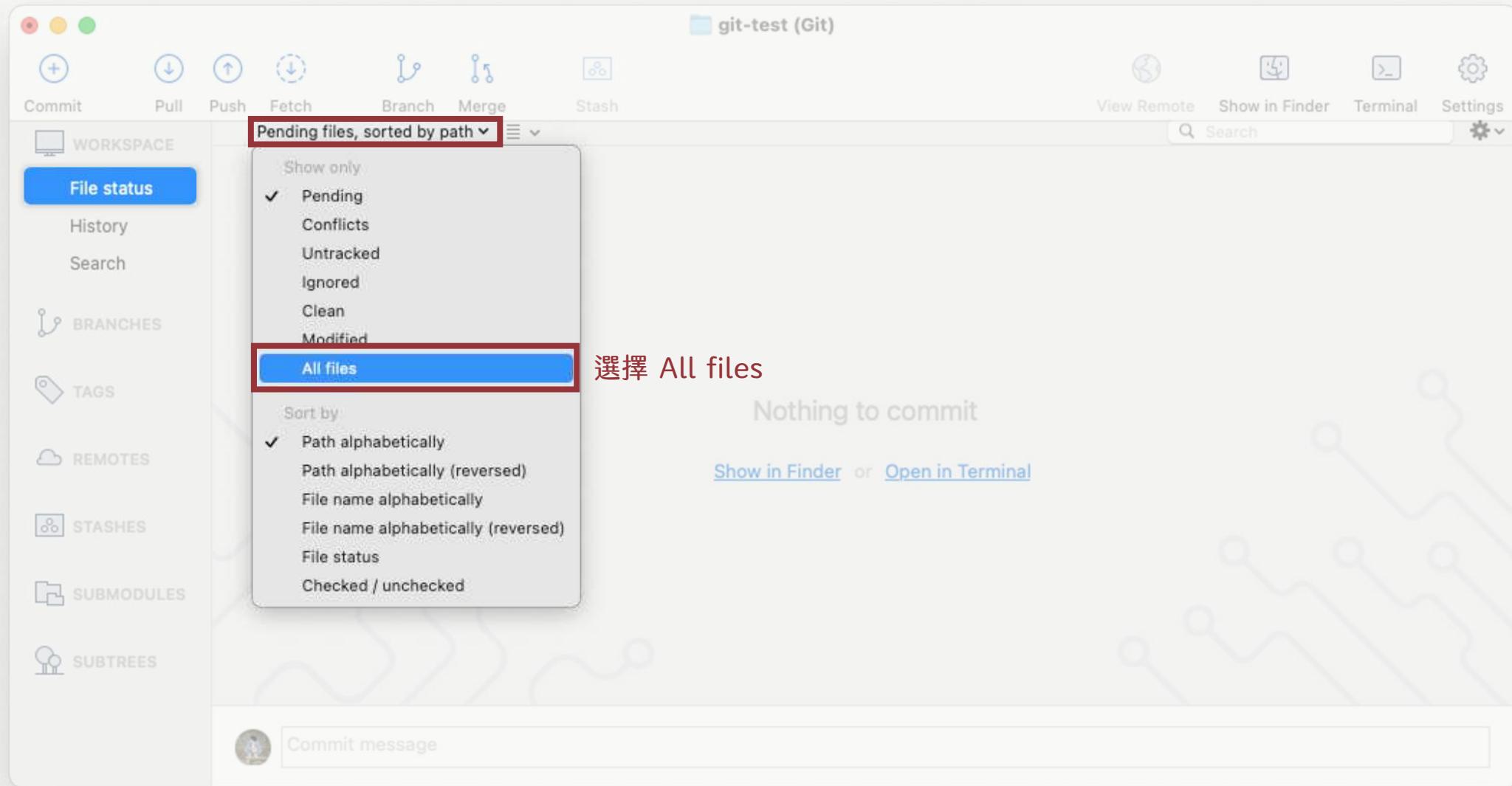
Sourcetree (log)



Sourcetree (blame)



Sourcetree (blame)



Sourcetree (blame)

The screenshot shows the Sourcetree application interface. On the left is a sidebar with navigation links: WORKSPACE (selected), File status, History, Search, BRANCHES, TAGS, REMOTES, STASHES, SUBMODULES, and SUBTREES. The main workspace shows two files: 'about.html' (selected and highlighted with a blue border) and 'index.html'. A context menu is open over 'about.html', listing options like Open, Show In Finder, Copy Path To Clipboard, etc. At the bottom of the menu, the option 'Annotate Selected...' is highlighted with a red box and labeled '選擇 Annotate Selected'.

右鍵

Annotate Selected...

選擇 Annotate Selected

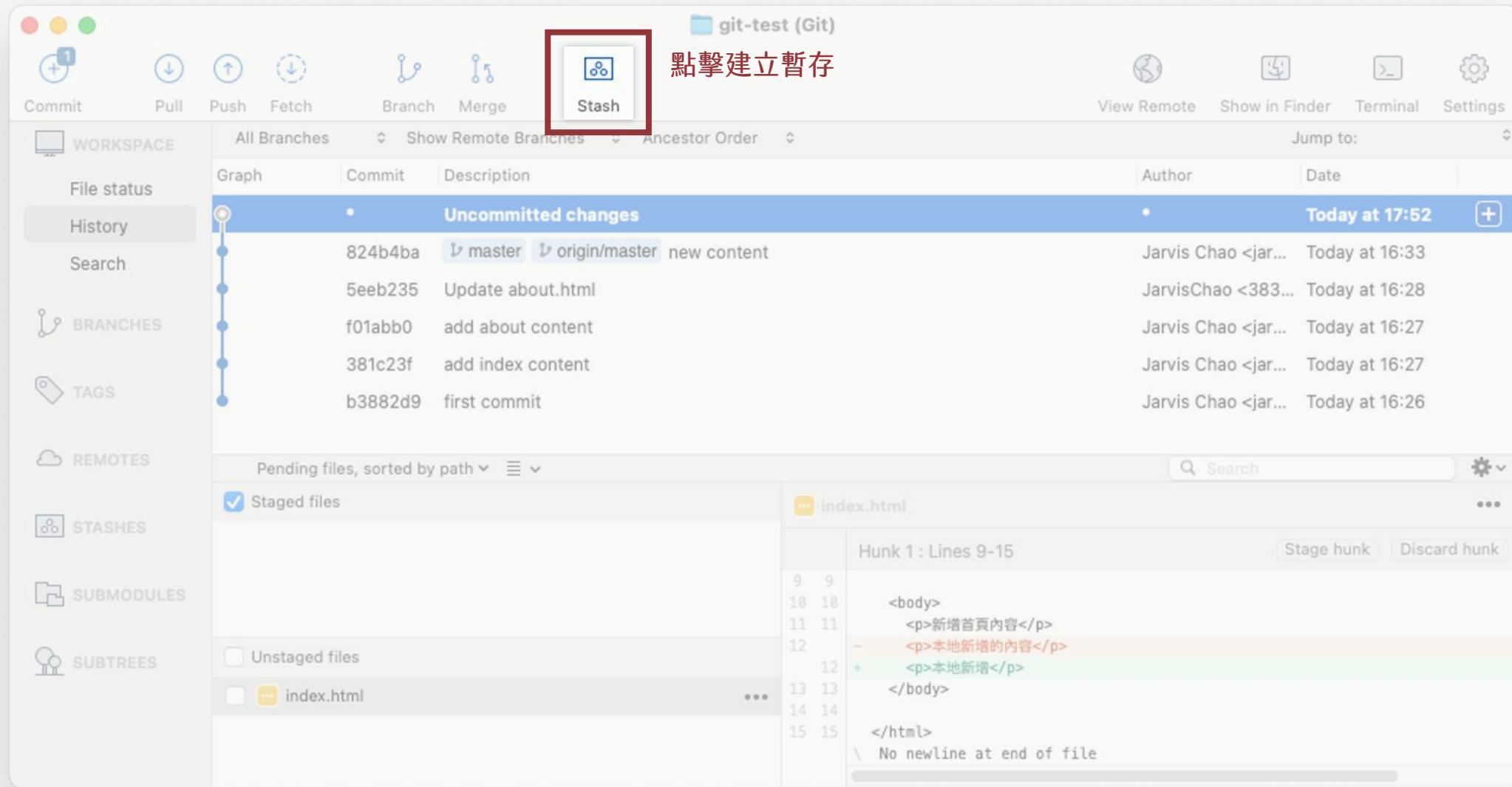
Sourcetree (blame)

The screenshot shows the Sourcetree application interface. On the left is a sidebar with icons for Commit, Pull, Workspace (selected), History, Search, Branches, Tags, Remotes, Stashes, Submodules, and Subtrees. The main window title is "Annotations for about.html". The content area displays a table with four columns: Line, Author, Changeset, and Content. The first row is highlighted with a blue background. The content of the table is as follows:

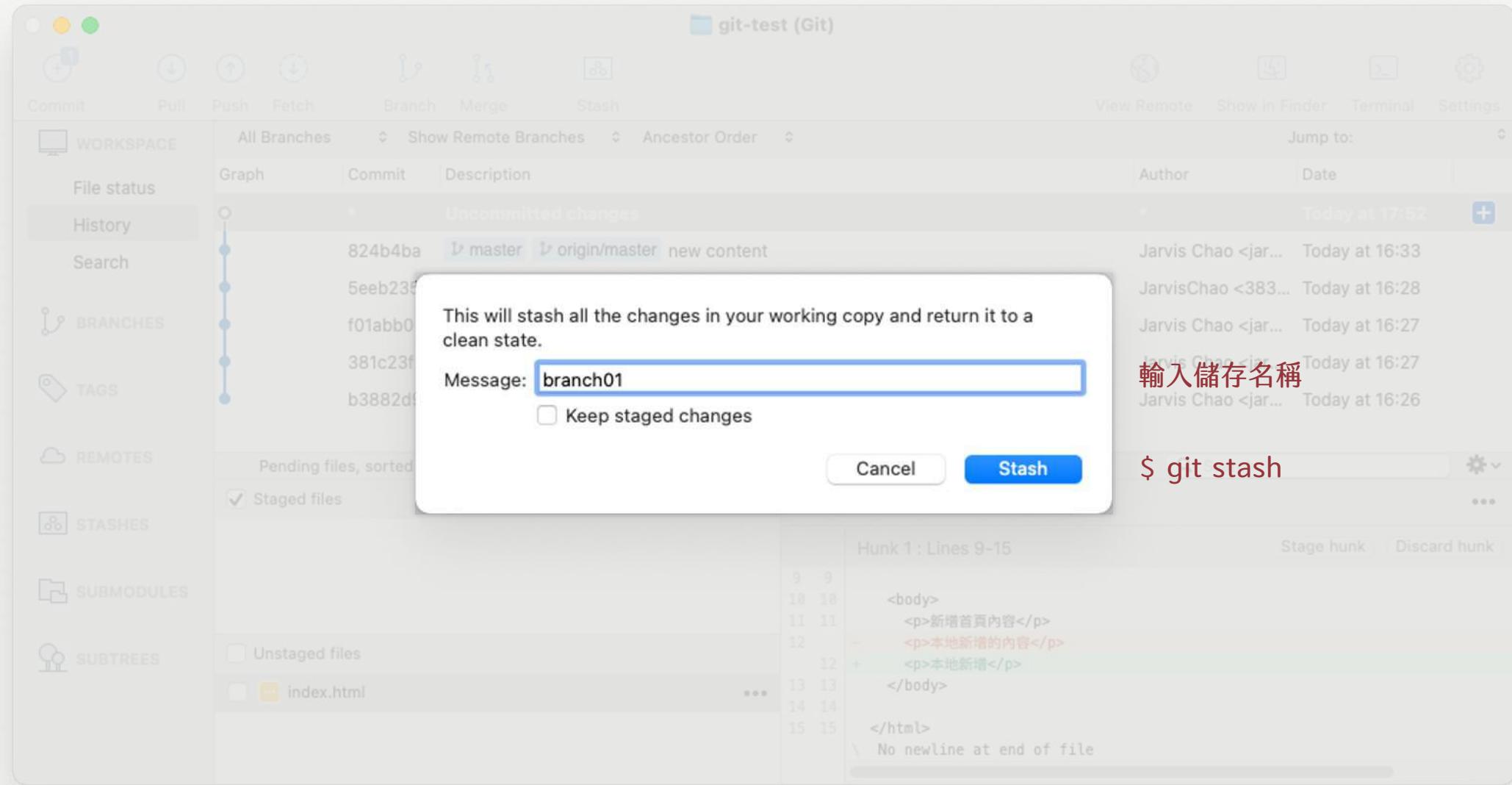
Line	Author	Changeset	Content
1	Jarvis Chao	2482e694	<!DOCTYPE html>
2	Jarvis Chao	2482e694	<html lang="en">
3	Jarvis Chao	2482e694	<head>
4	Jarvis Chao	2482e694	<meta charset="UTF-8">
5	Jarvis Chao	2482e694	<meta name="viewport" content="width=device-width, initial-scale=1.0">
6	Jarvis Chao	2482e694	<meta http-equiv="X-UA-Compatible" content="ie=edge">
7	Jarvis Chao	2482e694	<title>Document</title>
8	Jarvis Chao	2482e694	</head>
9	Jarvis Chao	2482e694	
10	Jarvis Chao	2482e694	<body>
11	Jarvis Chao	2482e694	<!-- // -->
12	Jarvis Chao	2482e694	</body>
13	Jarvis Chao	2482e694	
14	Jarvis Chao	2482e694	</html>

A vertical scrollbar is visible on the right side of the content area. A "Close" button is located at the bottom right of the modal window.

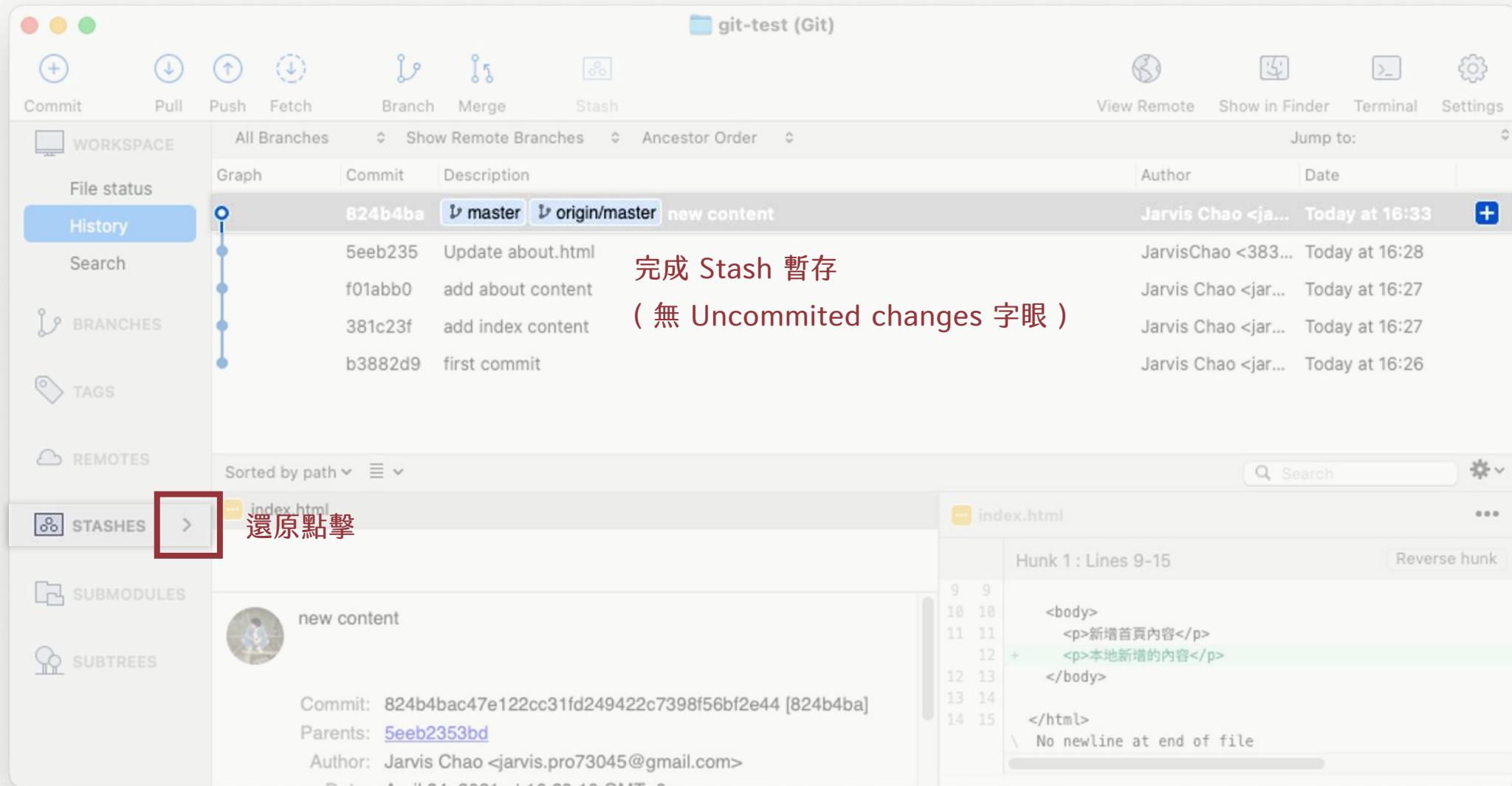
Sourcetree (stash)



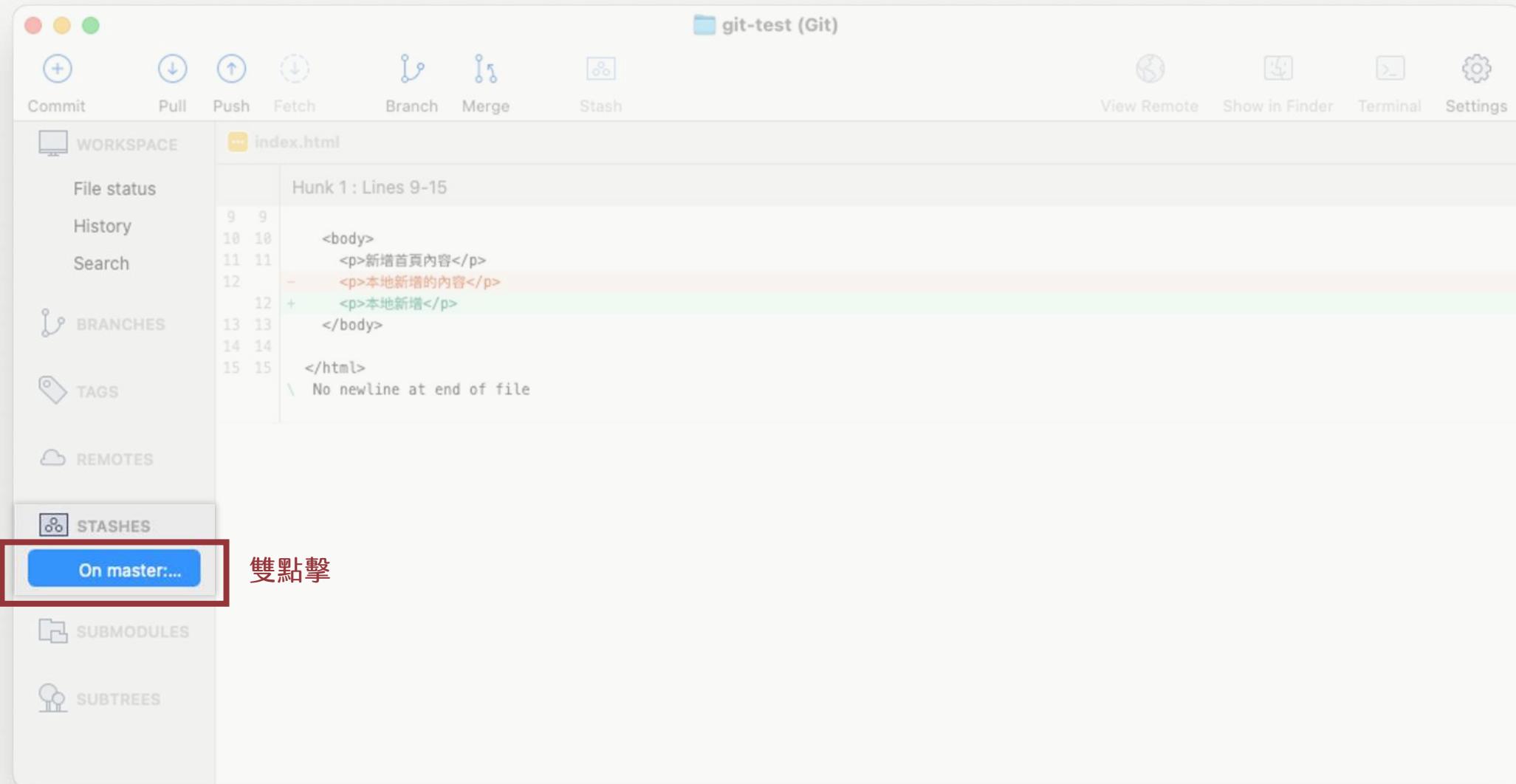
Sourcetree (stash)



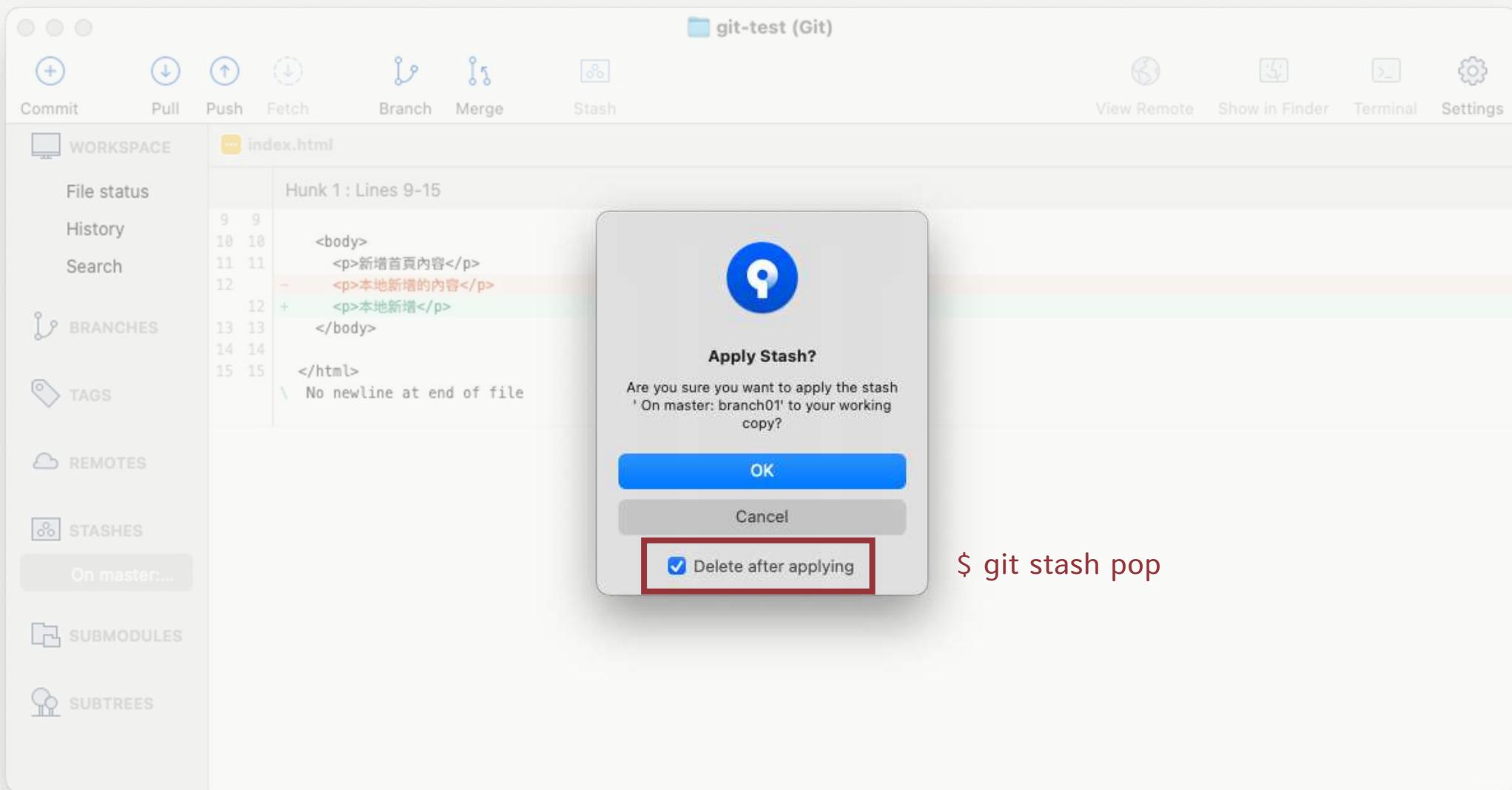
Sourcetree (stash)



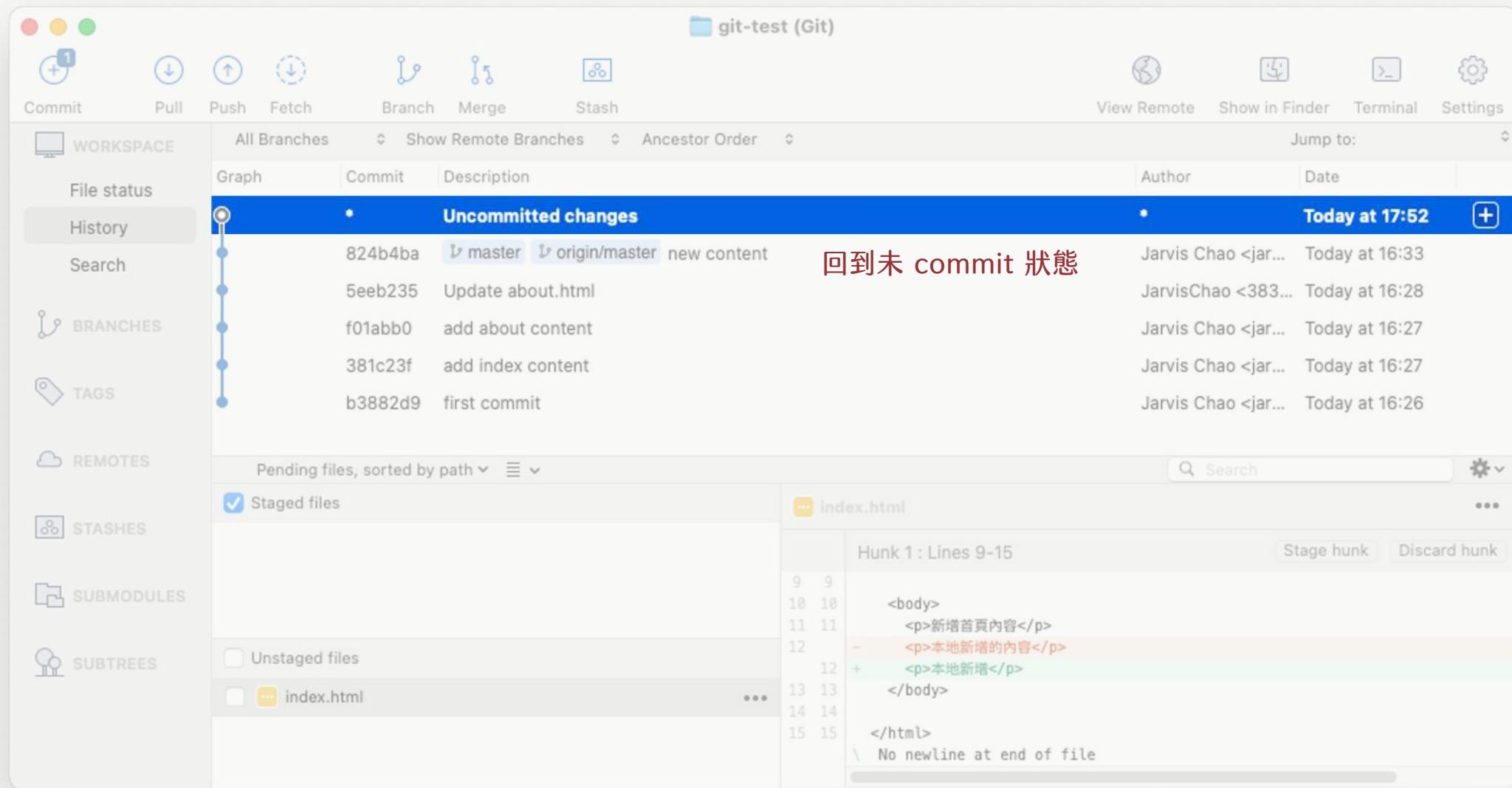
Sourcetree (stash)



Sourcetree (stash)



Sourcetree (stash)



Git 分支及合併

使用 branch 開分支並使用 merge 合併

分支及合併

\$ git branch

在不動到原檔的情形下，另開分支新增功能

```
$ git branch  
$ git branch name // 新增分支
```

\$ git merge 分支名

先切回 master 分支再合併其他分支到 master 分支

```
$ git checkout master  
$ git merge branch01
```

\$ git checkout 分支名

切換 HEAD 至指定分支，也可查看指定版本內容

```
$ git checkout branch01 // 切換主
```

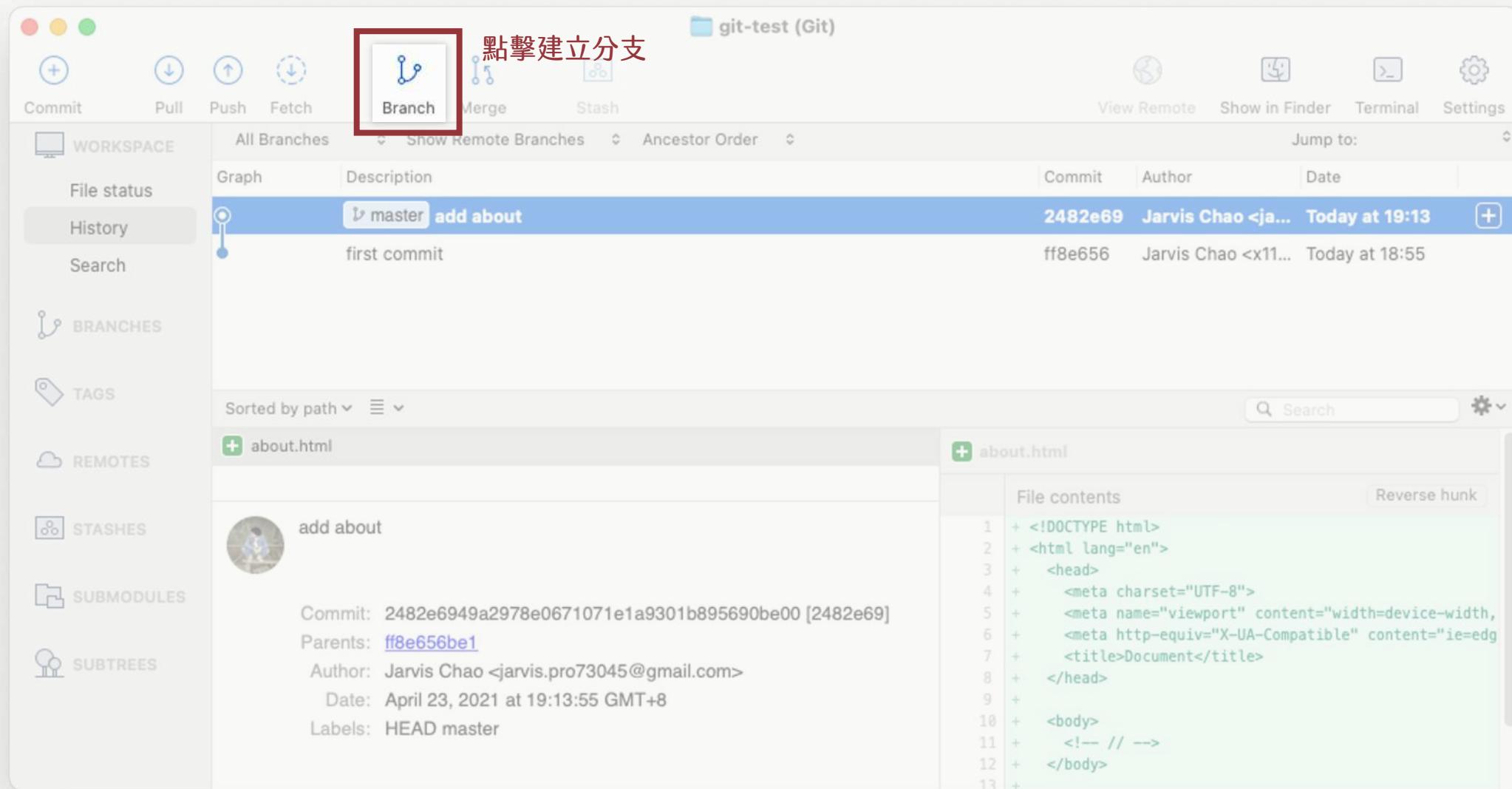
\$ git rebase 分支名

先切至所選分支，再將其合併成基礎

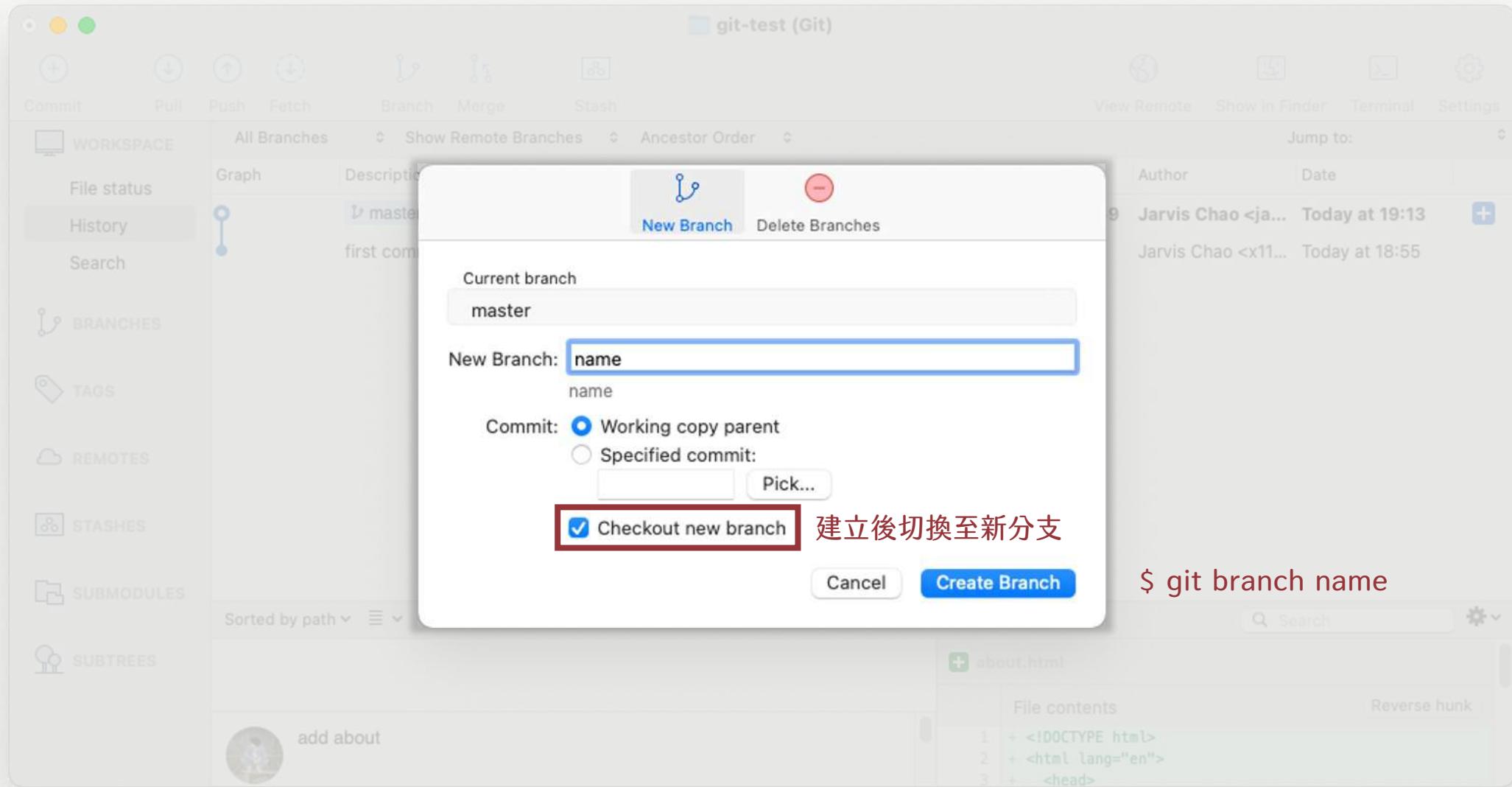
```
$ git checkout branch02  
$ git rebase branch01
```

TIPS : HEAD 為目前所在分支，可使用 checkout 控制位置；常用方法為：跳至指定版本並開分支重新從該版本修改

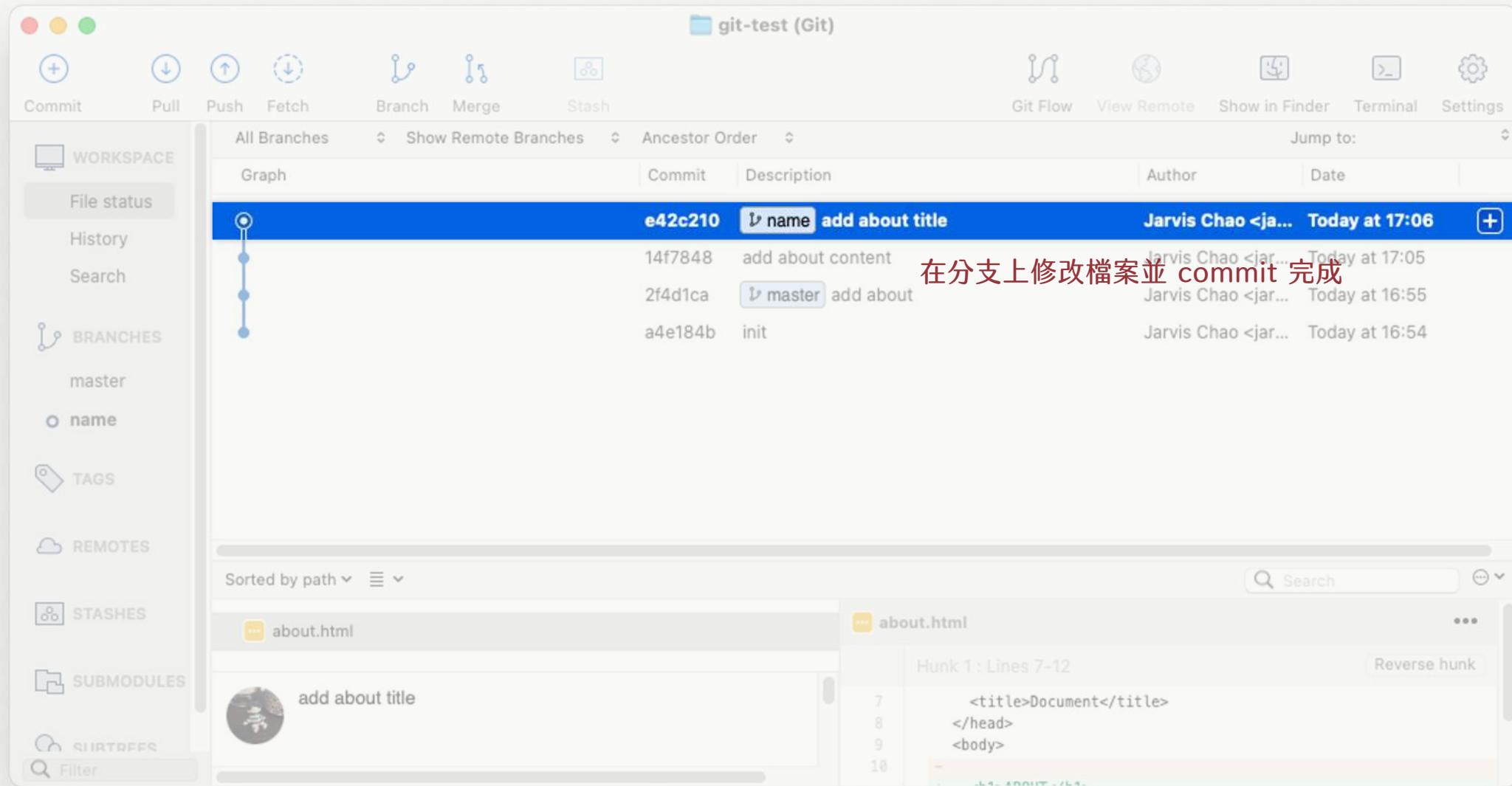
Sourcetree (branch)



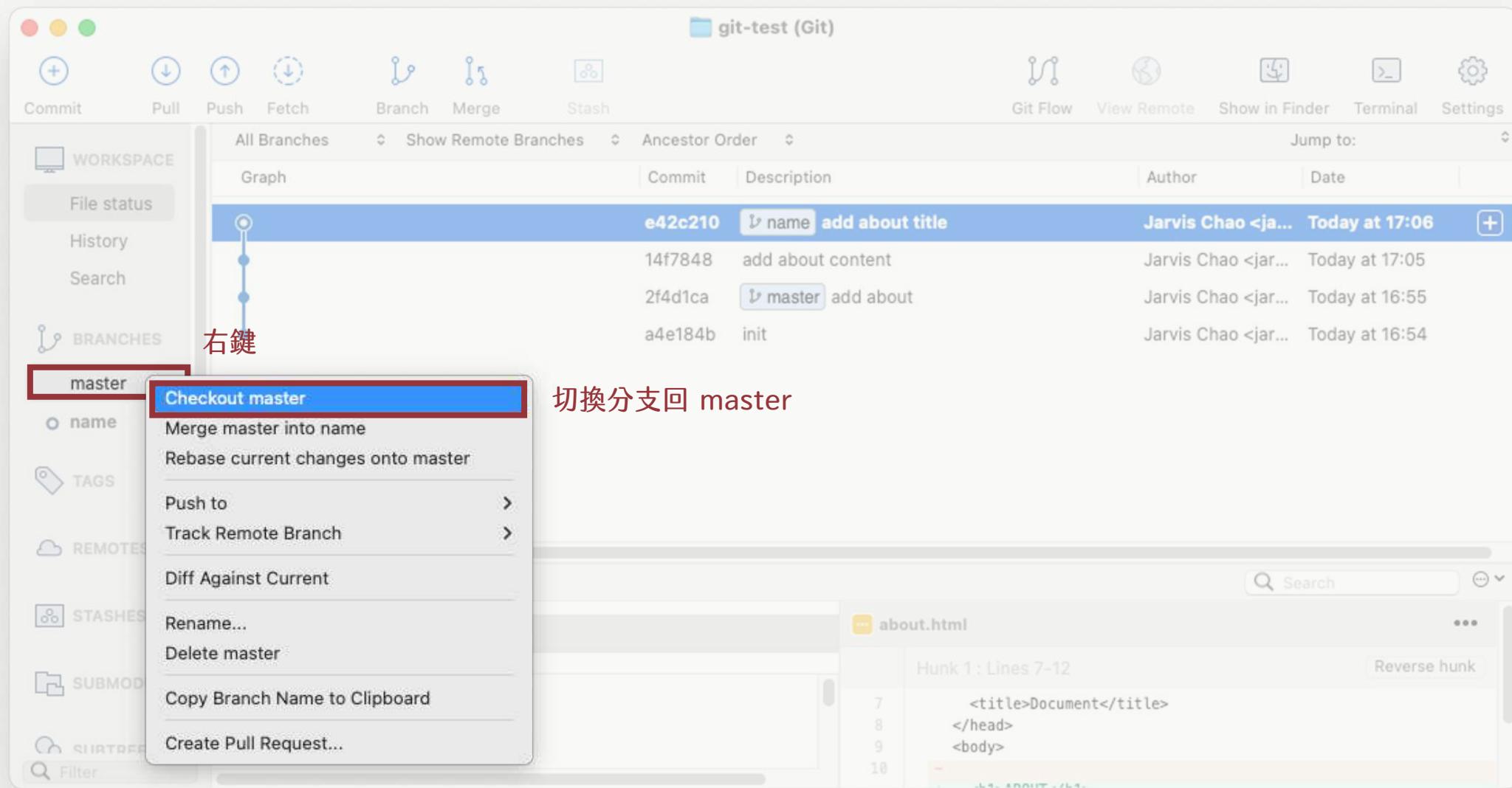
Sourcetree (branch)



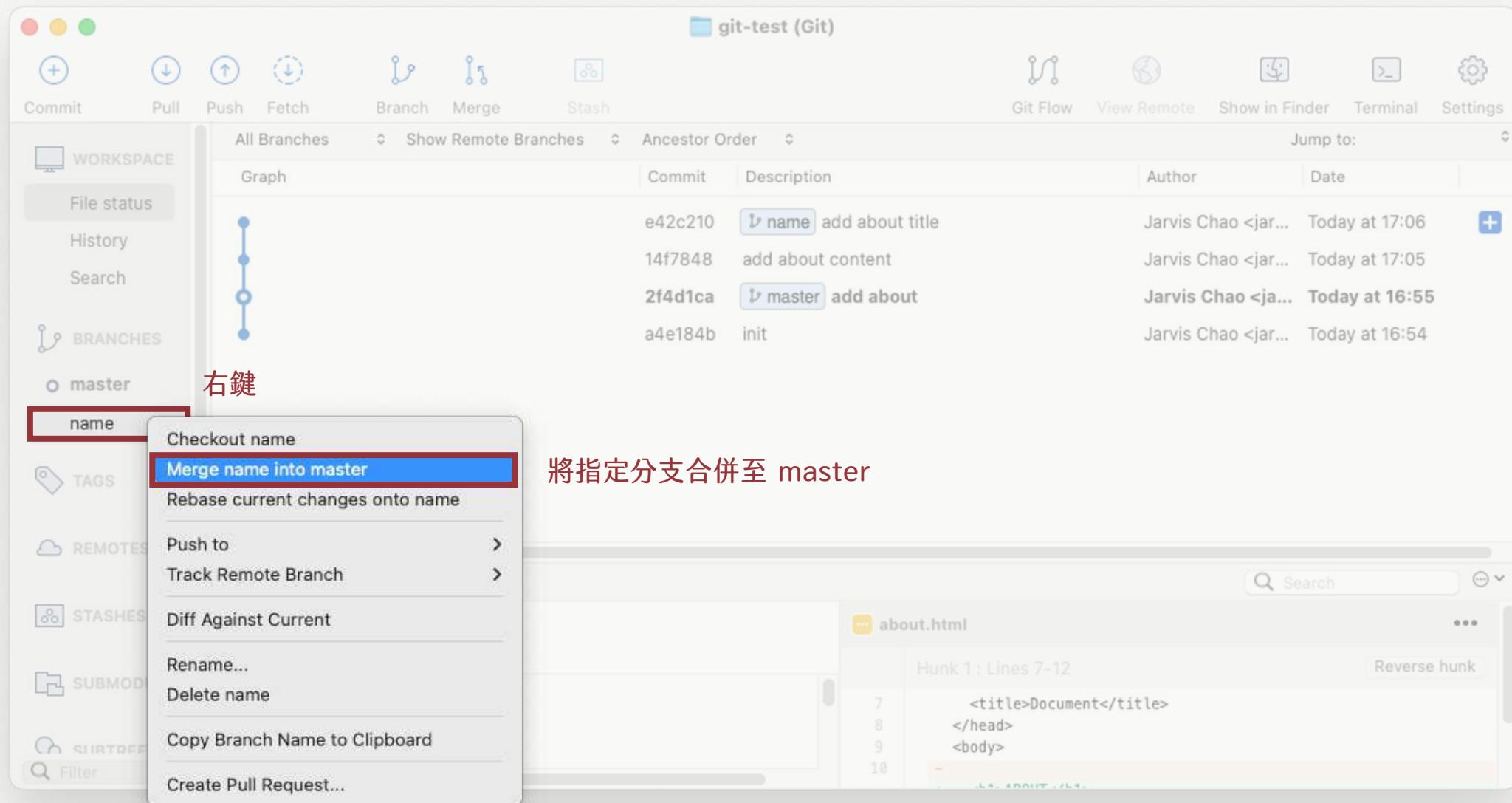
Sourcetree (branch)



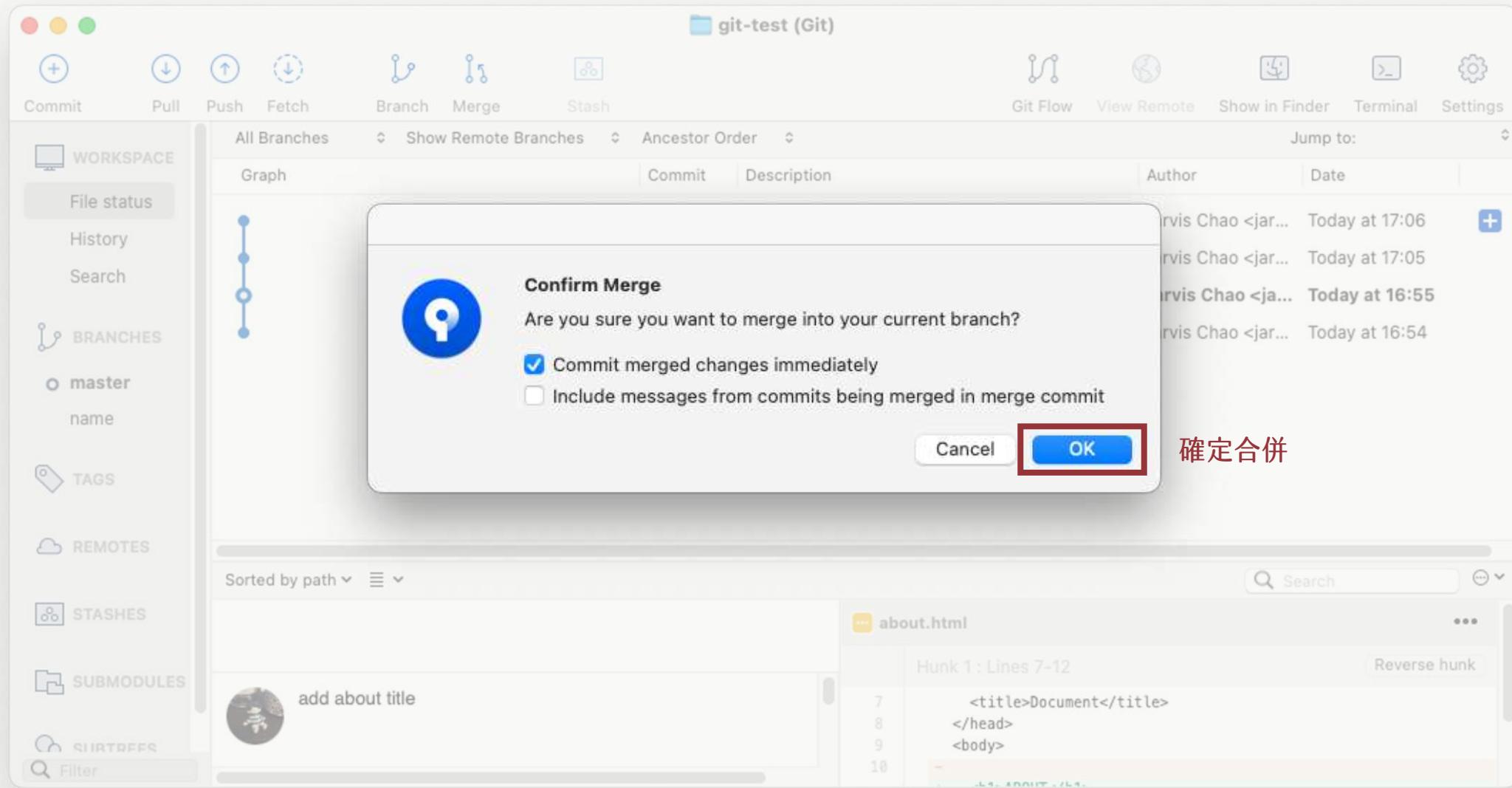
Sourcetree (checkout)



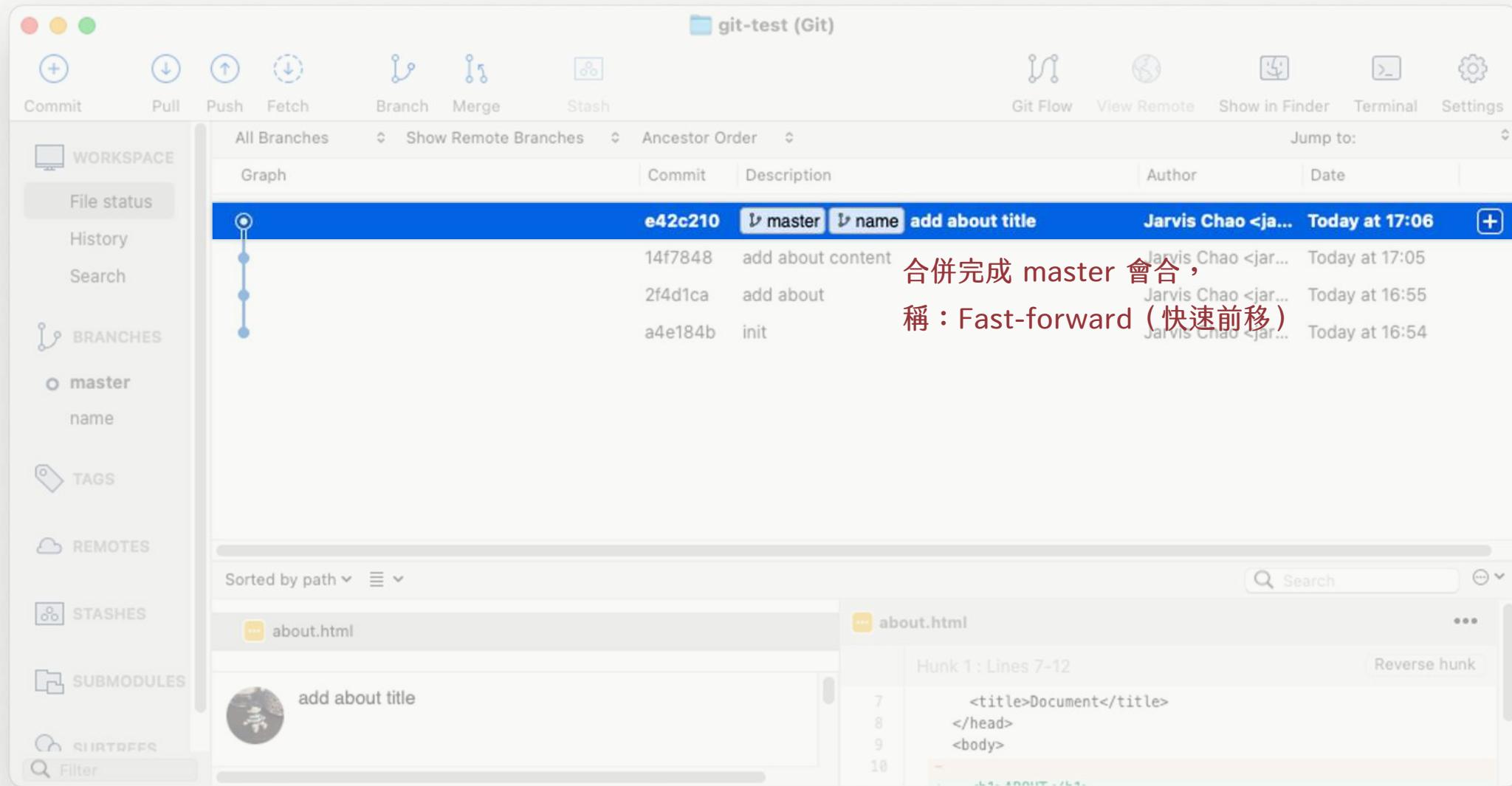
Sourcetree (merge)

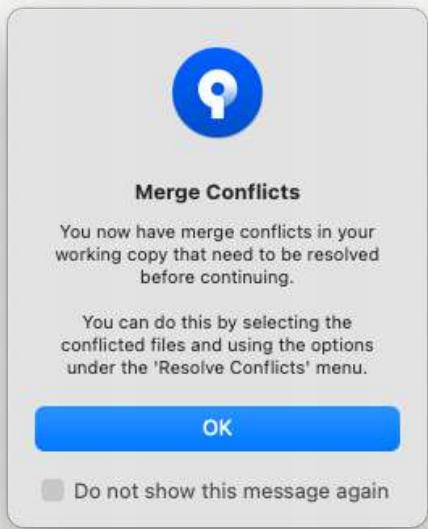


Sourcetree (merge)



Sourcetree (merge)





Merge 時發生衝突

Merge 時發生衝突

The screenshot shows a Microsoft Visual Studio Code (VS Code) interface with the following details:

- Explorer View:** Shows two open files: `index.html` and `about.html`. The `about.html` file is currently being edited.
- Editor View:** The file `about.html` contains an HTML document structure. A conflict is indicated between line 11 (from HEAD) and line 18 (from branch02). The conflict markers are:
 - Line 11: `<===== HEAD (Current Change)`
 - Line 12: `<p>reset前23</p>`
 - Line 13: `=====`
 - Line 14: `<p>新增about內容</p>`
 - Line 15: `<p>新增一些about內容</p>`
 - Line 16: `<p>branch2新增的about內容</p>`
 - Line 17: `<p>回到branch2新增內容</p>`
 - Line 18: `>>>>> branch02 (Incoming Change)`
 - Line 19: `</body>`
 - Line 20: `</html>`
- Status Bar:** Shows the current branch as `branch-back!`, and the status bar includes: Ln 20, Col 1, Spaces: 2, UTF-8, LF, HTML, Go Live, ESLint.

A red box highlights the text "手動解決衝突" (Manually resolve conflict) centered below the code editor.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta http-equiv="X-UA-Compatible" content="ie=edge">
<title>Document</title>
</head>
<body>
<===== HEAD (Current Change)
<p>reset前23</p>
=====
<p>新增about內容</p>
<p>新增一些about內容</p>
<p>branch2新增的about內容</p>
<p>回到branch2新增內容</p>
>>>>> branch02 (Incoming Change)
</body>
</html>
```

Git 回到指定版本

使用 `reset` 或 `checkout` 回到指定版本

回到指定版本

\$ git reset 指定編號

重置檔案至指定版本，如不熟悉有可能無法再 push 至遠端

```
$ git reset f2cbd99  
// 把檔案丟回工作目錄（預設）  
$ git reset f2cbd99 --mixed  
// 把檔案丟回暫存區域  
$ git reset f2cbd99 --soft  
// 把檔案直接丟掉  
$ git reset f2cbd99 --hard
```

\$ git reflog

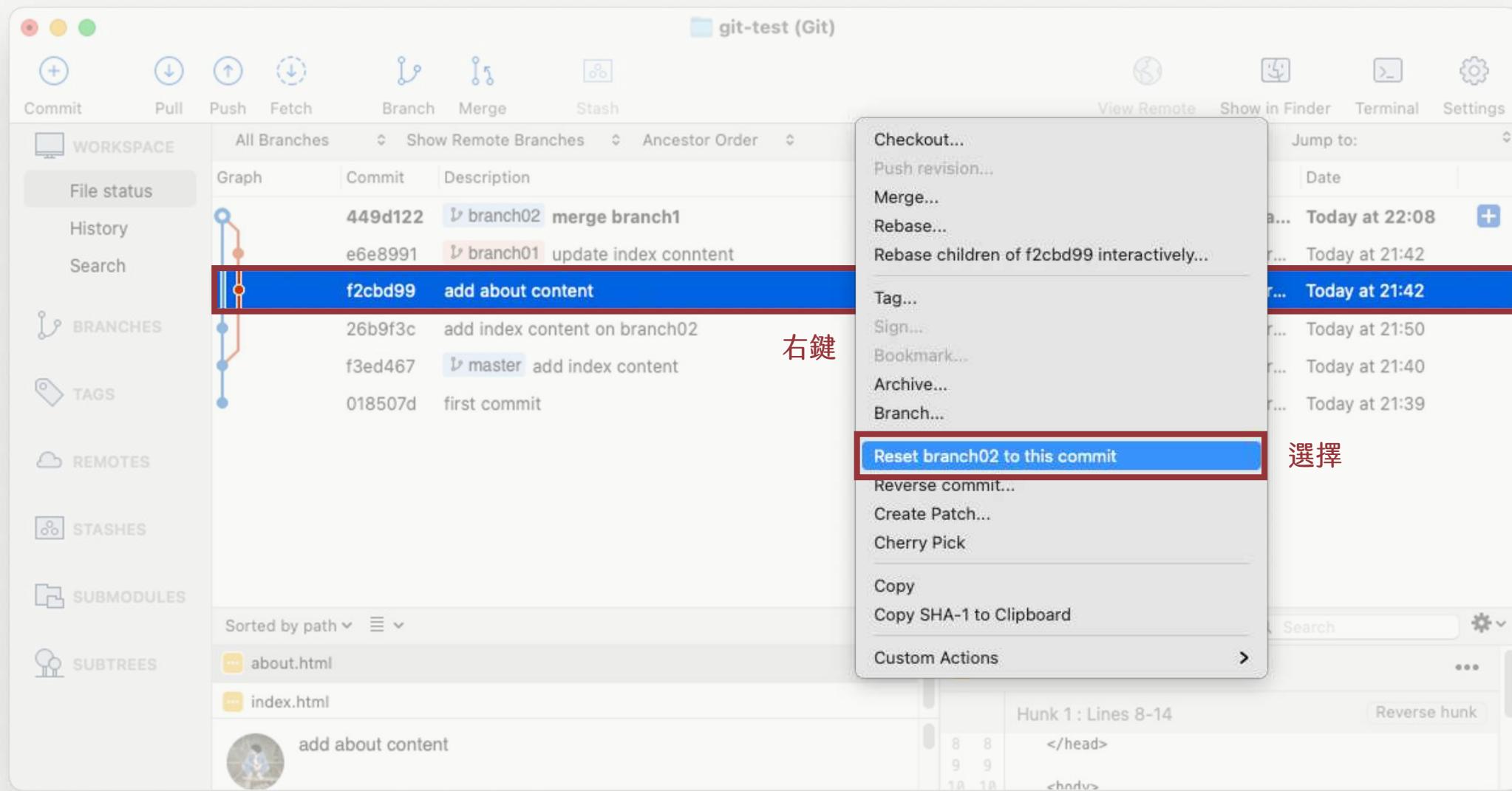
找出所有移動軌跡

```
$ git reflog  
$ :q // 離開 reflog  
  
// 找回使用 reset --hard 蓋掉的檔案  
$ git reset 449d122 --hard
```

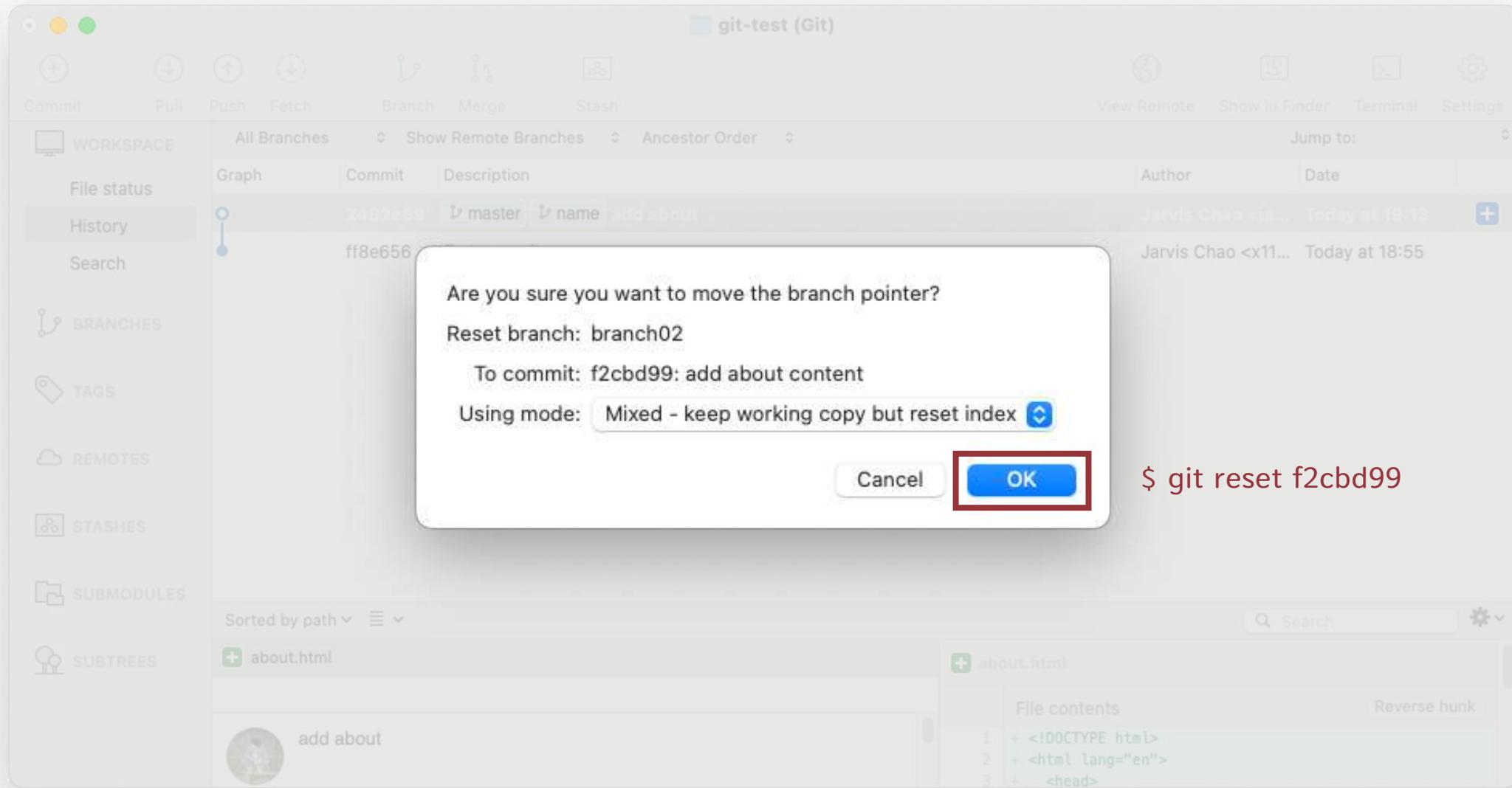
TIPS :

reset 盡量少用，一旦 commit 已經被 push 到遠端，那就不允許去 reset 本地資料；
建議使用 checkout 至指定版本後，再另開新分支編輯。

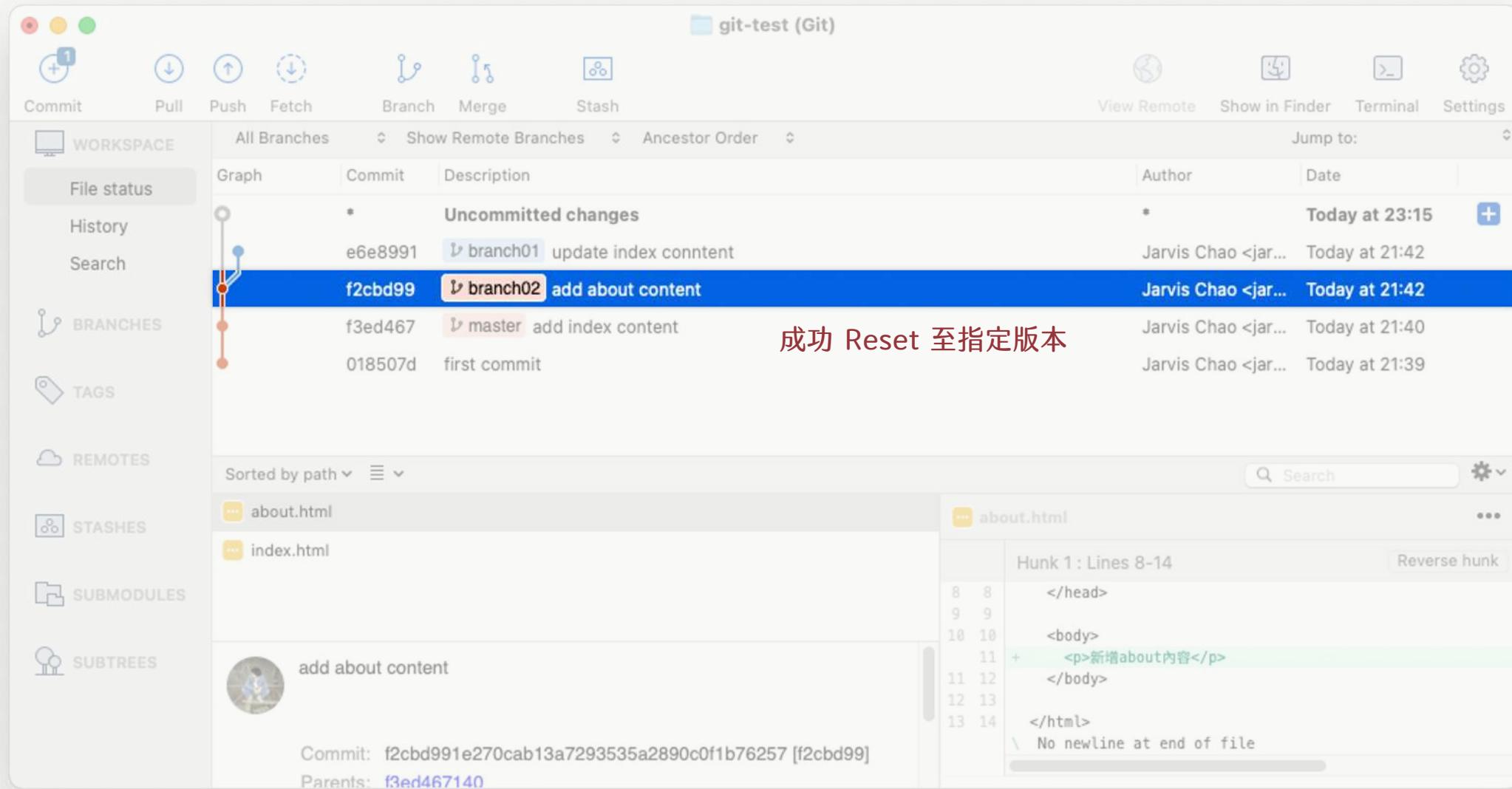
Sourcetree (reset)



Sourcetree (reset)



Sourcetree (reset)



Git 遠端資料傳輸

使用 push 與 pull 傳送及拉取遠端資料

遠端資料傳輸

\$ git remote add origin 遠端網址

設置遠端空間節點連線，宣告後續推拉檔案的地方

```
$ git remote add origin https://url/to/remote.git
```

\$ git push origin master

將資料推至遠端空間

```
$ git push origin master
```

\$ git pull

將資料從遠端拉取並合併 (merge)

```
$ git pull  
$ git pull --rebase // pull 後以 rebase 的方式合併
```

\$ git fetch

將資料從遠端拉取，不合併

```
$ git fetch origin master
```

遠端資料複製

\$ git clone 遠端網址

將遠端資料複製一份至本地資料夾（非下載，有完整 .git 檔）

```
$ git clone https://url/to/remote.git
```

取得 Personal access tokens

取得個人訪問 token 並在第三方平台使用

GitHub 取得 Personal access tokens

The screenshot shows a GitHub profile page for the user 'JarvisChao'. The page includes a sidebar with 'Top Repositories' (MEDIISCI/hunt.vue.app, JarvisChao/princity, etc.), 'Recent activity' (greensock/GSAP, JCloudYu, anewway), and 'Your teams' (MEDIISCI/case-dev). The main area displays activity feeds for 'Following' and 'For you (Beta)'. A prominent message says 'You've been added to the Princityle organization.' On the right, there's a 'Start coding instantly with GitHub Codespaces' section and a 'Latest changes' feed. A vertical dropdown menu on the right is signed in as 'JarvisChao' and lists options like 'Your profile', 'Your repositories', and 'Settings'. The 'Settings' option is highlighted with a red box.

Signed in as
JarvisChao

Search or jump to... Pull requests Issues Codespaces Marketplace Explore

You've been added to the Princityle organization. View organization

Following For you (Beta)

anewway pushed to MEDIISCI/hunt.vue.app · 2 days ago

1 commit to production cc43299 Fix earn color class digits to 4 instead of 2 for display consistency...

anewway pushed to MEDIISCI/hunt.vue.app · 2 days ago

1 commit to production 90e2764 Add URL_TRADING_VIEW_WEBHOOK_URL_TEACHING_MANUAL link button.

JCloudYu pushed to MEDIISCI/hunt.vue.app · last week

1 commit to production a611a6c Locale text improvements

anewway pushed to MEDIISCI/hunt.vue.app · last week

2 commits to production ca522db Hide CoverPos button in follow or signal open mode in trade status page. e058d9b Remove has_purchased check logic in custom settings page.

Start coding instantly with GitHub Codespaces

Spin up fully configured dev environments on powerful VMs that start in seconds. Get up to 60 hours a month of free time.

Get started

Latest changes

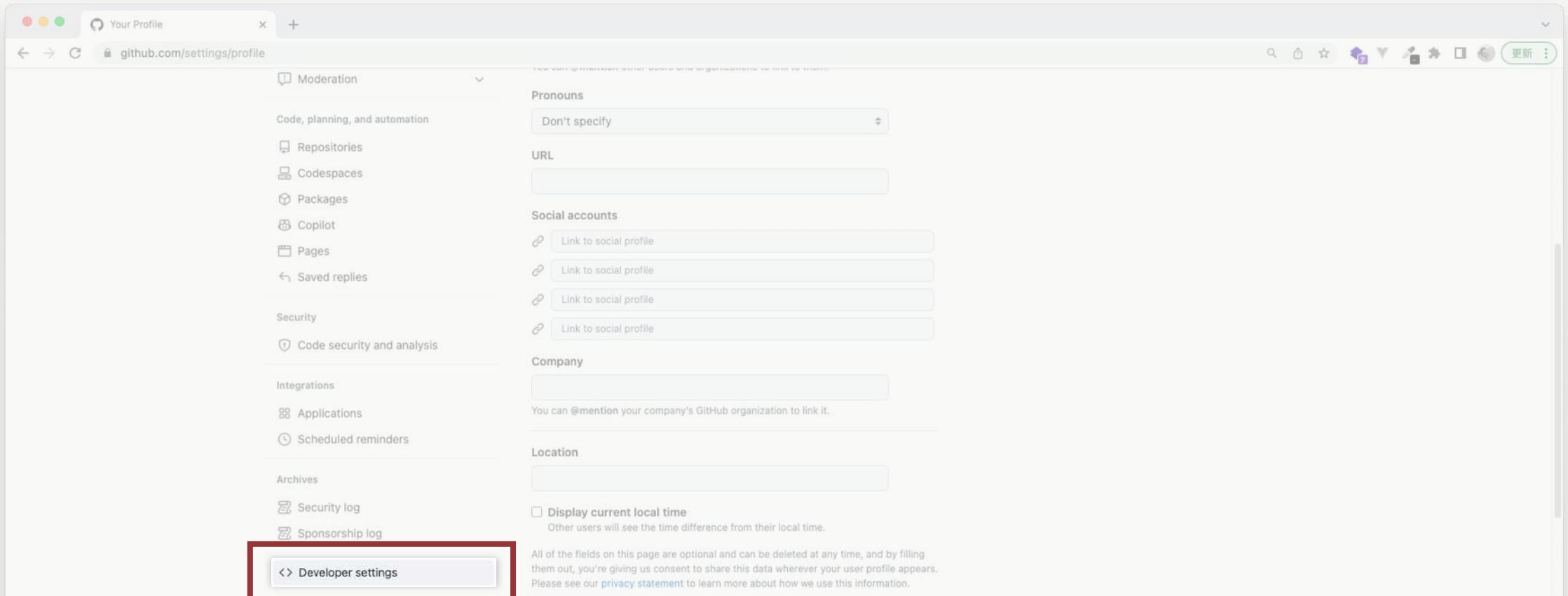
- 12 hours ago SMS and TOTP can now both be registered
- 12 hours ago Accessibility improvements for the content
- 12 hours ago GitHub Issues & Projects – March 2nd update
- Yesterday GitHub Actions: Admins can now bypass rules

View changelog →

Upgrade Try Enterprise Feature preview Help Settings Sign out

<https://github.com/settings/profile>

GitHub 取得 Personal access tokens



The screenshot shows the GitHub 'Your Profile' settings page. On the left, there's a sidebar with various profile management sections like Moderation, Code, planning, and automation, Security, Integrations, Archives, and Applications. The 'Developer settings' button at the bottom of this sidebar is highlighted with a red box. The main right-hand area contains fields for Pronouns (set to 'Don't specify'), URL, Social accounts (with four 'Link to social profile' buttons), Company (with a placeholder for @mentioning a company), Location, and a checkbox for 'Display current local time'. Below these is a note about optional fields being deletable. At the bottom is a green 'Update profile' button.

<https://github.com/settings/apps>

Contributions & Activity

Make profile private and hide activity
Enabling this will hide your contributions and activity from your GitHub profile and from social features like followers, stars, feeds, leaderboards and releases.

Include private contributions on my profile
Your contribution graph, achievements, and activity overview will show your private contributions without revealing any repository or organization information. Learn more

GitHub 取得 Personal access tokens

The screenshot shows the GitHub 'Personal Access Tokens (classic)' page. A red box highlights the 'Tokens (classic)' option under the 'Personal access tokens' dropdown menu. Another red box highlights the 'Generate new token (classic)' button, which is described as 'For general use'.

Personal access tokens (classic)

Generate new token (classic)

選擇 Tokens (classic)

https://github.com/settings/tokens/new

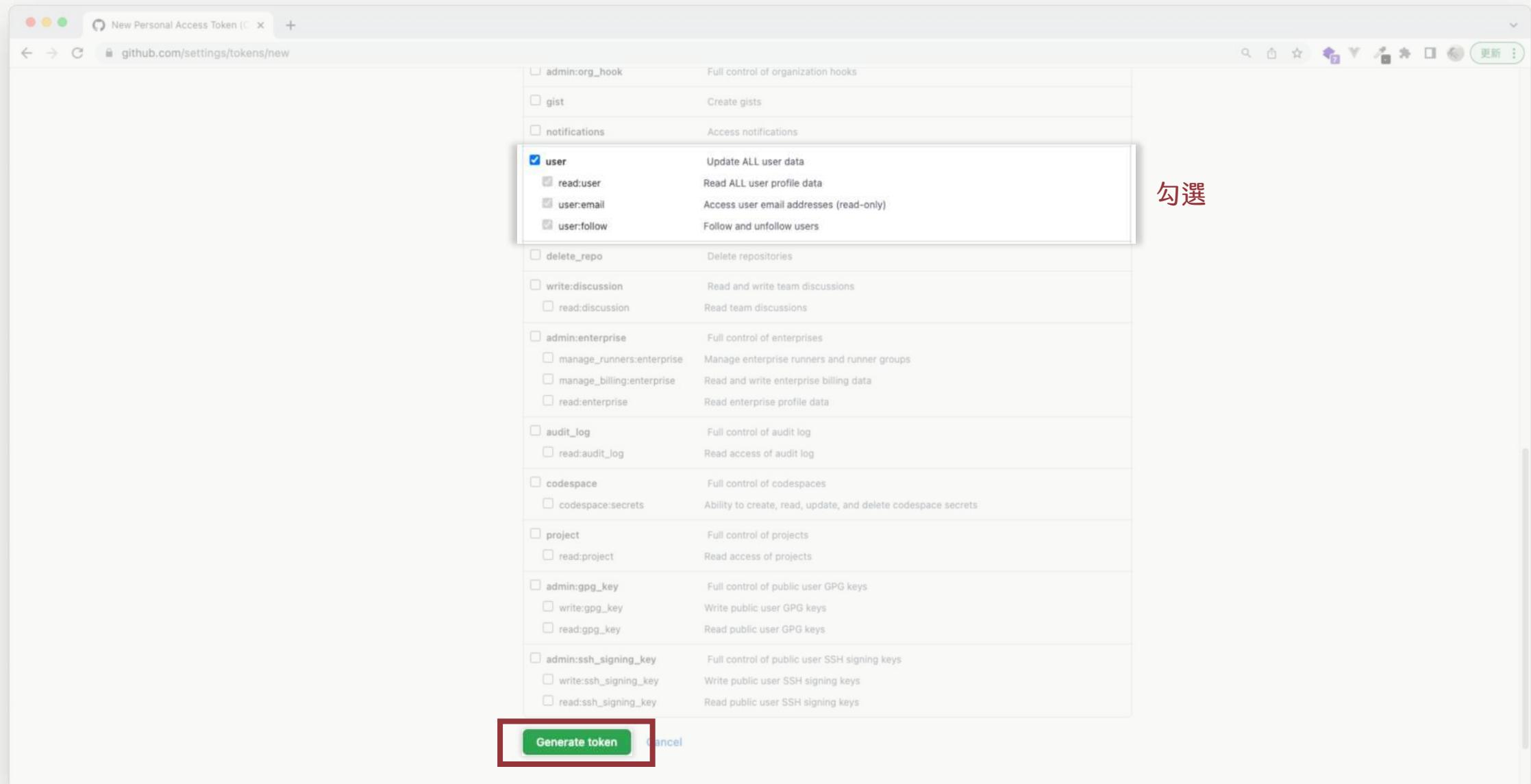
GitHub 取得 Personal access tokens

The screenshot shows a browser window for creating a new personal access token on GitHub. The URL is github.com/settings/tokens/new. The left sidebar has sections for GitHub Apps, OAuth Apps, Personal access tokens (selected), Fine-grained tokens (Beta), and Tokens (classic). The main area is titled "New personal access token (classic)". It includes a note about what the token is for ("Git Push"), an expiration date set to "2023/04/04", and a "Select scopes" section. The "repo" scope is checked, granting full control of private repositories. Other scopes listed include repo:status, repo_deployment, public_repo, repo:invite, security_events, workflow, write:packages, read:packages, delete:packages, admin:org, write:org, and read:org.

輸入 Title 及時效

勾選

GitHub 取得 Personal access tokens



GitHub 取得 Personal access tokens

The screenshot shows the GitHub 'Personal Access Tokens (classic)' page. On the left sidebar, under 'Personal access tokens', the 'Tokens (classic)' option is selected. The main area displays a table of tokens:

Token	Scopes	Last used	Action
ghp_Ar5lPl60DRQjjuzAeHNP0Aw2Y5dYke42pIaI	Git Deploy — repo, user	Expires on Thu, Jul 27 2023.	Delete
Git Push — repo, user		Last used within the last week	Delete

A red box highlights the first token row. A red annotation '保存起來' (Save) is placed to the right of the highlighted row. A note at the bottom states: 'Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.'

Settings / Developer settings

Personal access tokens (classic)

Generate new token ▾ Revoke all

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

Make sure to copy your personal access token now. You won't be able to see it again!

ghp_Ar5lPl60DRQjjuzAeHNP0Aw2Y5dYke42pIaI

Git Deploy — repo, user

Expires on Thu, Jul 27 2023.

Git Push — repo, user

Last used within the last week

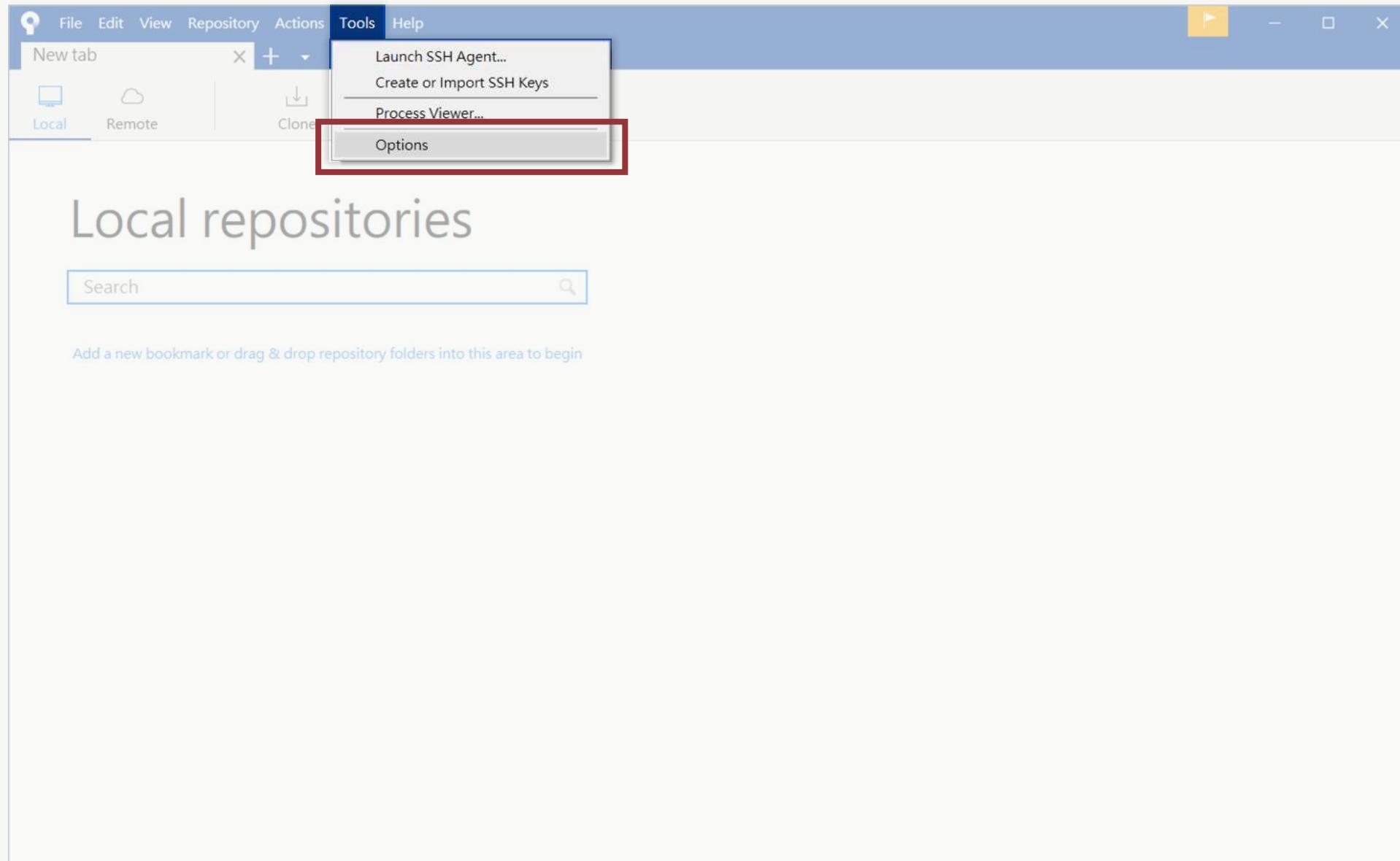
Deletes

© 2023 GitHub, Inc. Terms Privacy Security Status Docs Contact GitHub Pricing API Training Blog About

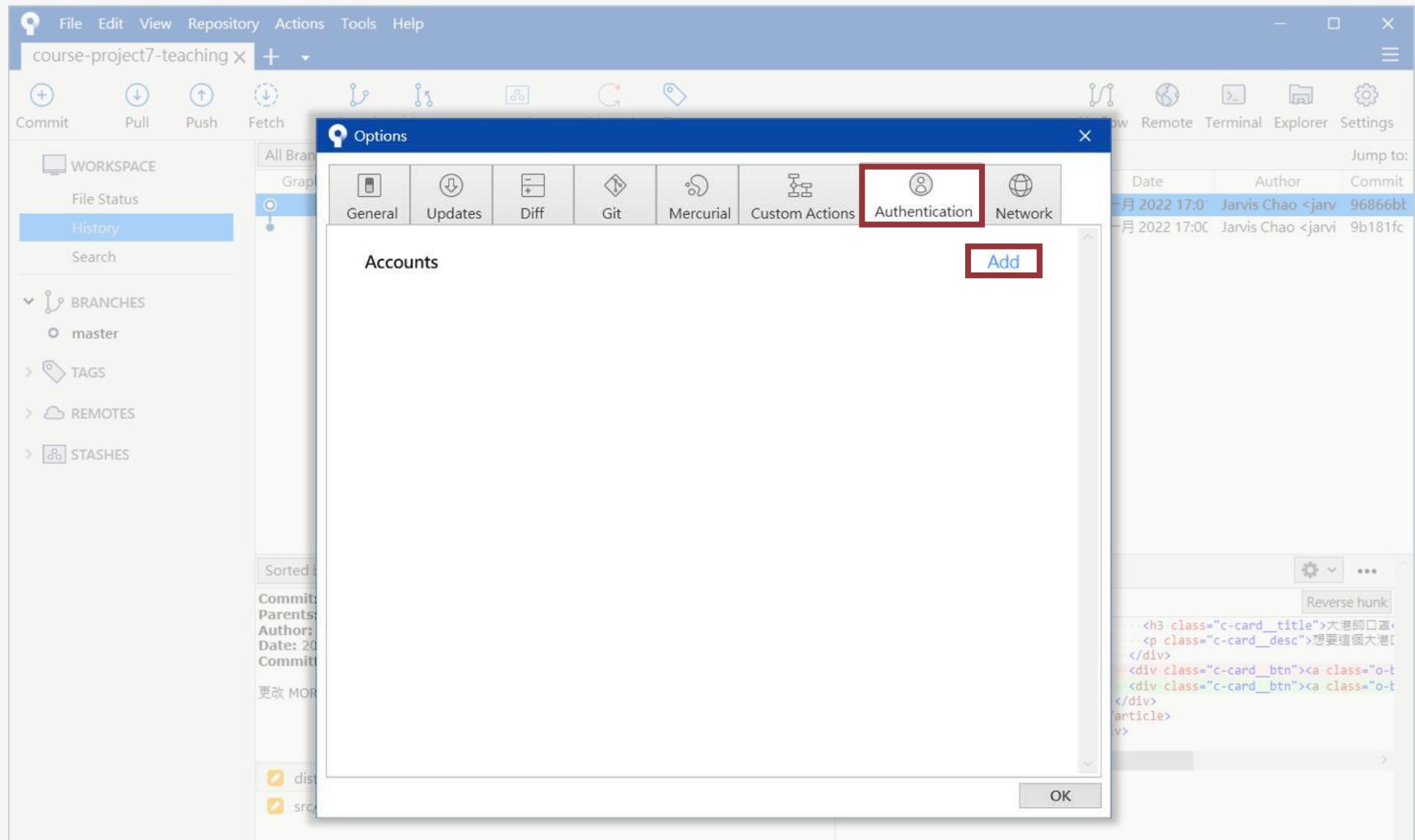
Sourcetree 連結 Github 帳戶

連結帳戶以便後續操作

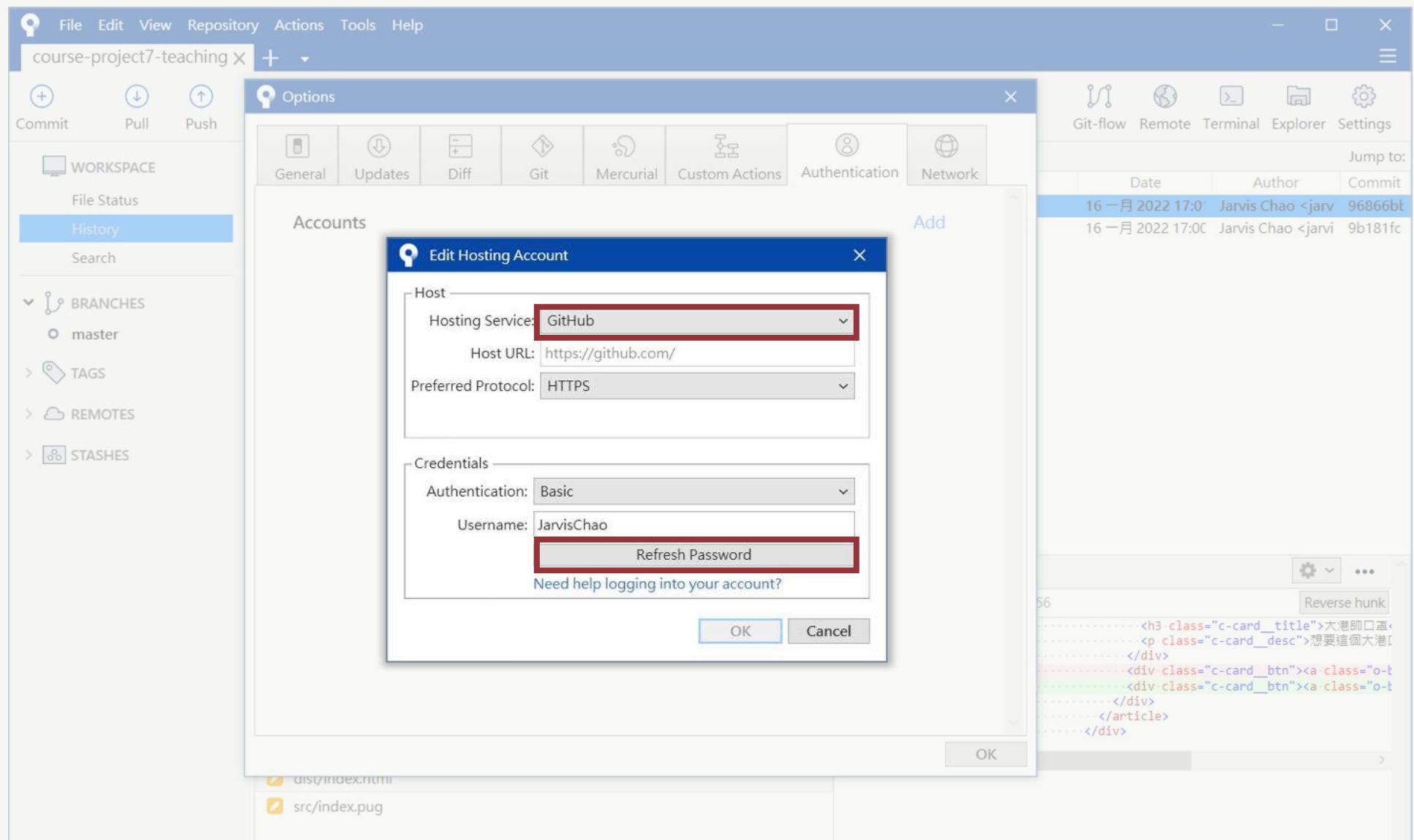
Sourcetree (連結 Github 帳戶)



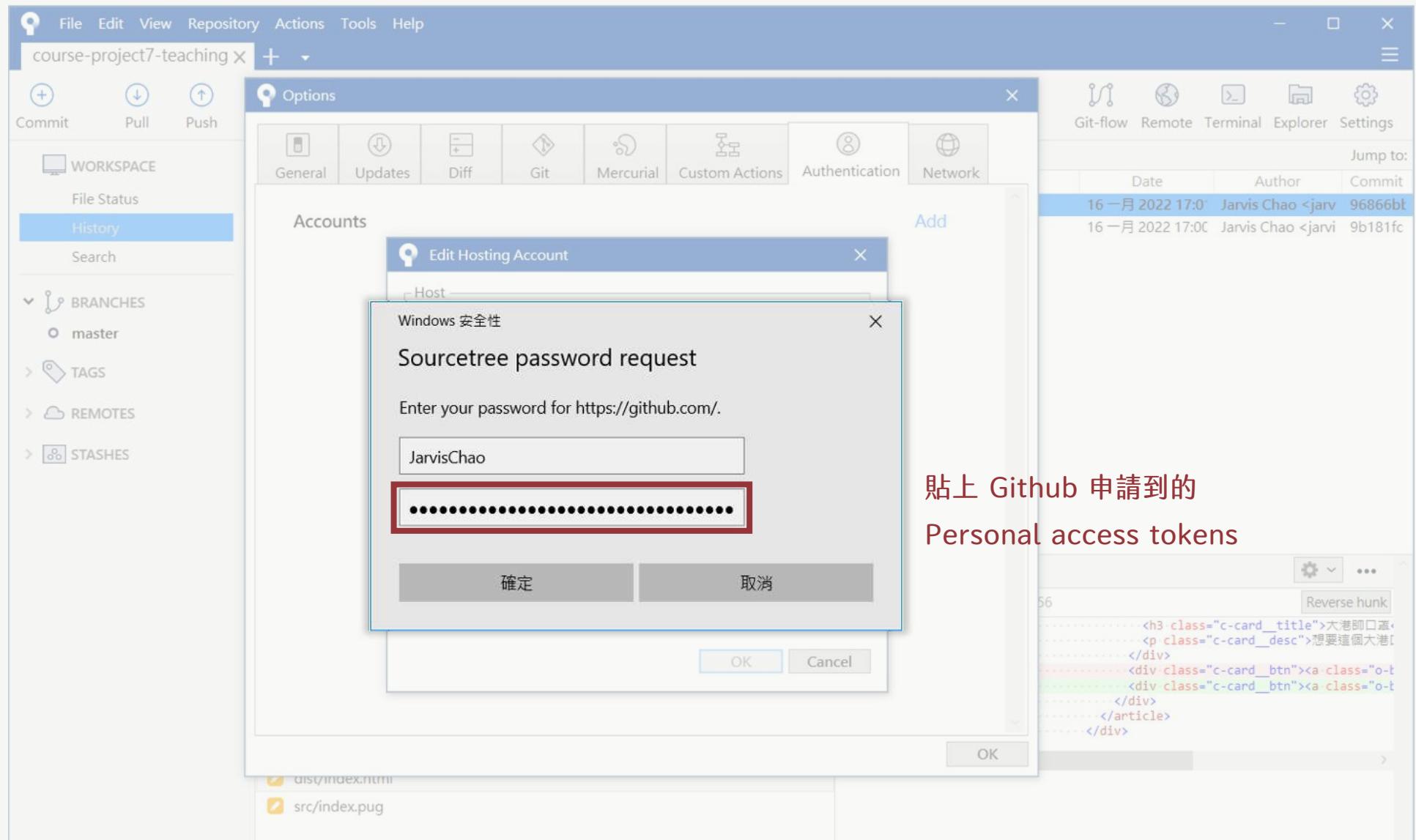
Sourcetree (連結 Github 帳戶)



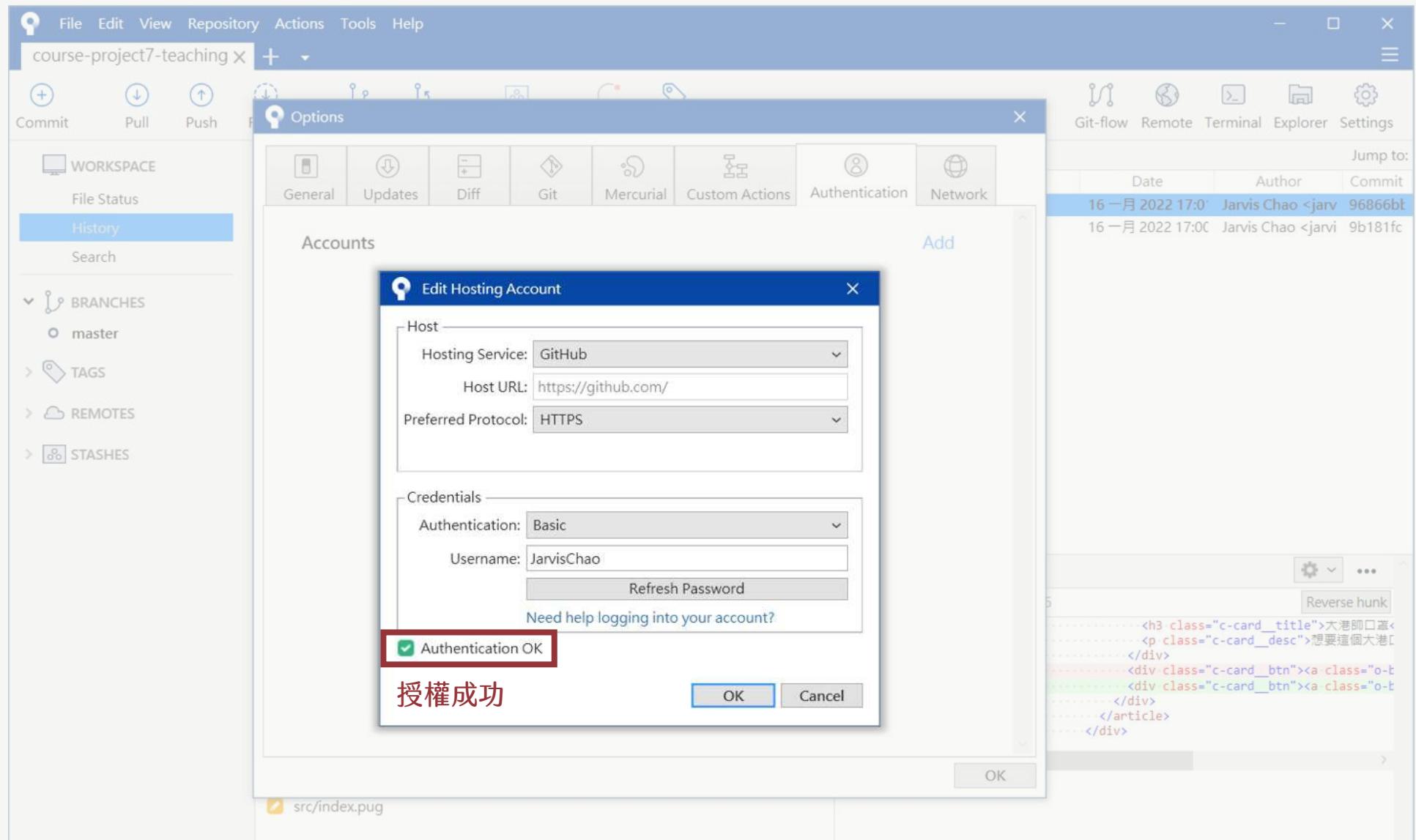
Sourcetree (連結 Github 帳戶)



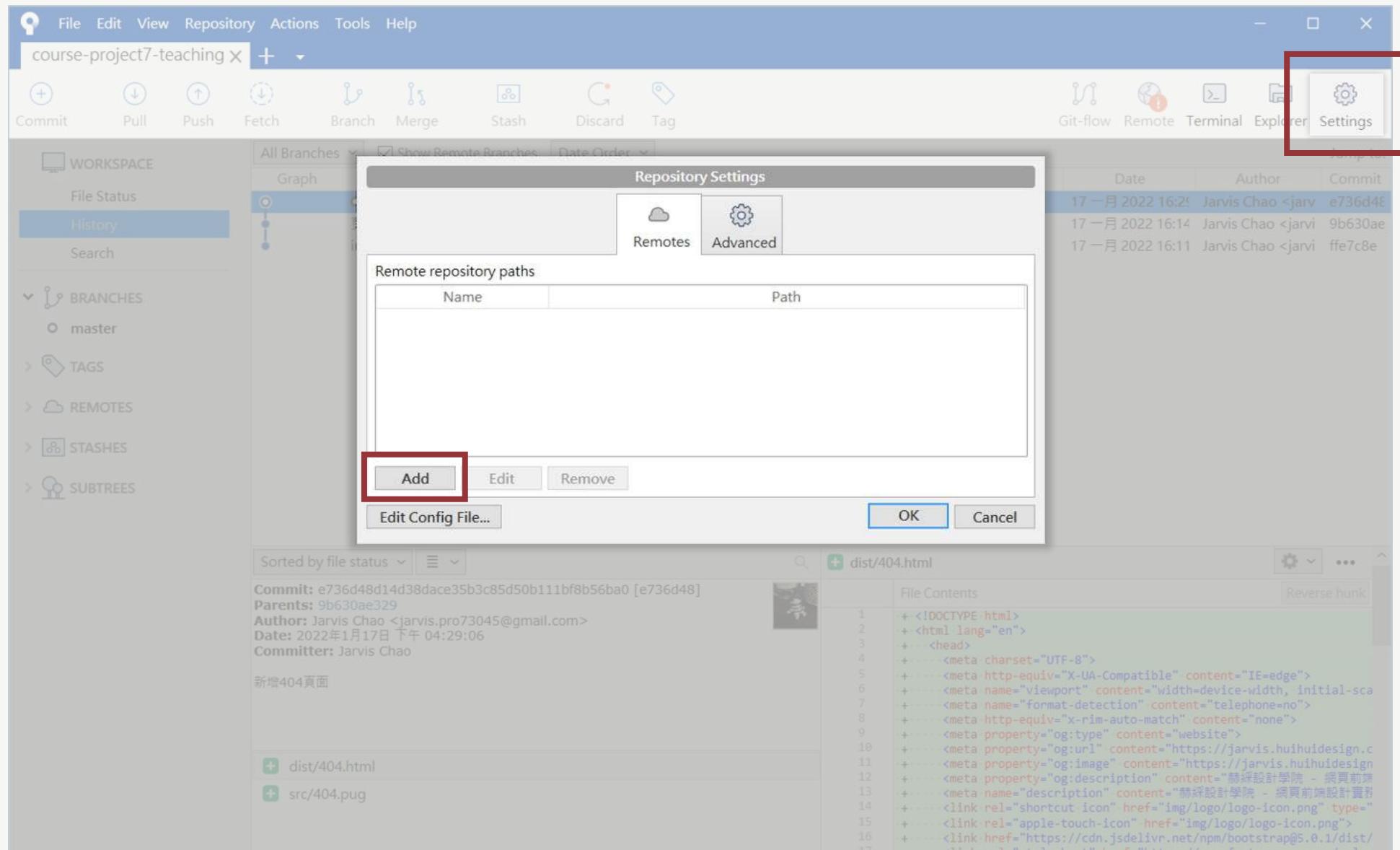
Sourcetree (連結 Github 帳戶)



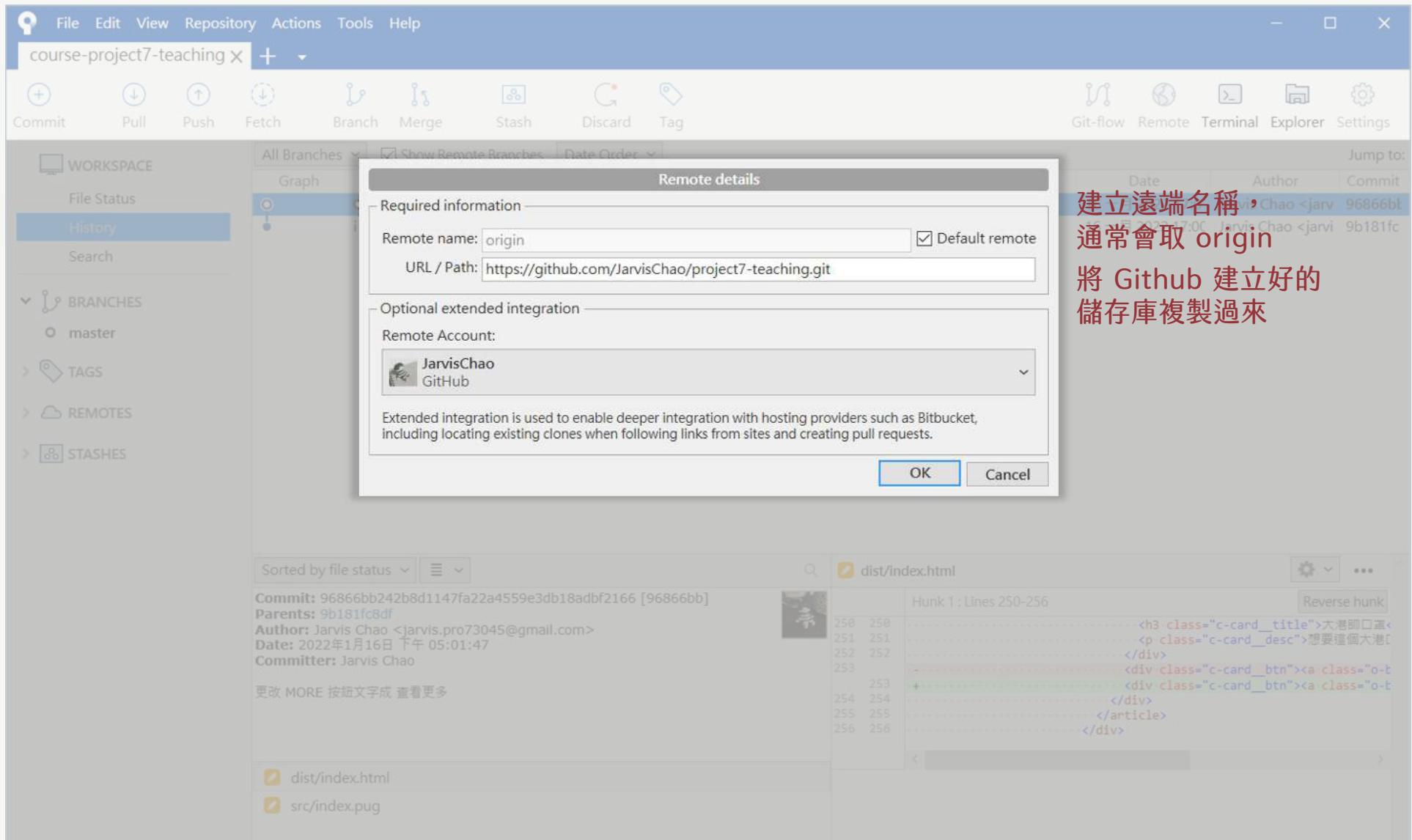
Sourcetree (連結 Github 帳戶)



Sourcetree (連結 Github 遠端庫)



Sourcetree (連結 Github 遠端庫)



建立 Github 遠端儲存庫

建立 Github Repositories 遠端儲存庫並連結至 Sourcetree

GitHub 建立遠端庫

The screenshot shows a GitHub profile page for the user 'JarvisChao'. The main content area displays recent activity, including pushes to the 'hunt.vue.app' repository by users 'anewway' and 'JCloudYu'. A prominent message通知 states: 'You've been added to the Princityle organization.' On the right side, a sidebar menu is open, with the 'Your repositories' option highlighted with a red box.

Signed in as
JarvisChao

View organization

Latest changes

- 12 hours ago SMS and TOTP can now both be registered
- 13 hours ago Accessible navigation bar for the current organization
- 13 hours ago GitHub Issues & Projects – March 2nd update
- Yesterday GitHub Actions: Admins can now bypass rules

View changelog →

開啟儲存庫

Your repositories

Your organizations Your projects Your stars Your gists Your sponsors

Upgrade Try Enterprise Feature preview Help Settings Sign out

Top Repositories

Find a repository...

MEDIISCI/hunt.vue.app

JarvisChao/princityle

JarvisChao/tanley

Sbl-lemaxim/ntcunionInsuranceWeb

JarvisChao/green-dev

Princityle/ochre

Sbl-lemaxim/Labor-Union-master

Show more

Recent activity

greensock/GSAP

Parallax effect jitter on mobile with ScrollTrigger

Your teams

Find a team...

MEDIISCI/case-dev

You've been added to the Princityle organization.

Following For you Beta

anewway pushed to MEDIISCI/hunt.vue.app - 2 days ago

1 commit to production

cc43299 Fix earn color class digits to 4 instead of 2 for display consistency...

anewway pushed to MEDIISCI/hunt.vue.app - 2 days ago

1 commit to production

98e2764 Add URL_TRADE_VIEW_WEBHOOK_URL_TEACHING_MANUAL link button.

JCloudYu pushed to MEDIISCI/hunt.vue.app - last week

1 commit to production

a611a6c Locale text improvements

anewway pushed to MEDIISCI/hunt.vue.app - last week

2 commits to production

ca522db Hide CoverPos button in follow or signal open mode in trade status page.

e058d9b Remove has_purchased check logic in custom settings page.

GitHub 建立遠端庫

The screenshot shows a GitHub user profile page with the URL github.com/JarvisChao?tab=repositories. The user has 14 repositories. A red box highlights the green 'New' button in the top right corner of the repository list header. To the right of the button, the text '建立新儲存庫' (Create New Repository) is displayed.

Repository list:

- megaport-dev** Public
赫採設計學院 - 網頁前端設計實務 | 整合應用 (Project7) , 建立開發環境並實作完整互動式網站。
Pug Updated 2 days ago
- princityle** Private
則式官網
Vue Updated last week
- tanley** Private
丹利居飾購物車網站
Vue Updated last week
- green-dev** Private
綠壯健累前台開發
Vue Updated 2 weeks ago
- khh-labor-union-dev** Private
Vue Updated on Jan 6

GitHub 建立遠端庫

The screenshot shows the GitHub 'Create a New Repository' interface. At the top, the URL is `github.com/new`. The main title is 'Create a new repository'. Below it, a sub-instruction says 'A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.' The 'Owner' dropdown is set to 'JarvisChao'. The 'Repository name *' field is highlighted with a red border and contains 'my-projects'. To the right of this field, the text '輸入儲存庫名稱' (Input repository name) is overlaid in red. Below the name field, a note says 'Great repository names are short and memorable. Need inspiration? How about shiny-octo-fishstick?' The 'Description (optional)' field contains '我的作品集專案'. Under 'Visibility', the 'Public' radio button is selected, with the note 'Anyone on the internet can see this repository. You choose who can commit.' To its right, the text '勾選 Public 才能使用部署功能' (Check Public to use deployment features) is overlaid in red. Below the visibility section, there's a note 'Initialize this repository with: Skip this step if you're importing an existing repository.' followed by an unchecked checkbox for 'Add a README file'. Further down, there's a note 'Choose which files not to track from a list of templates.' followed by a dropdown menu set to '.gitignore template: None'. At the bottom, there's a note 'Choose a license' with a dropdown menu set to 'License: None'. A small note at the very bottom left says 'ⓘ You are creating a public repository in your personal account.' At the bottom center, a large green button with white text says 'Create repository', which is also outlined with a red border.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner * JarvisChao

Repository name * my-projects 輸入儲存庫名稱

Great repository names are short and memorable. Need inspiration? How about shiny-octo-fishstick?

Description (optional)

我的作品集專案

Public
Anyone on the internet can see this repository. You choose who can commit.

Private
You choose who can see and commit to this repository.

勾選 Public 才能使用部署功能

Initialize this repository with:
Skip this step if you're importing an existing repository.

Add a README file
This is where you can write a long description for your project. Learn more.

Add .gitignore
Choose which files not to track from a list of templates. Learn more.

.gitignore template: None

Choose a license
A license tells others what they can and can't do with your code. Learn more.

License: None

ⓘ You are creating a public repository in your personal account.

Create repository

GitHub 建立遠端庫

The screenshot shows a web browser window with the GitHub URL `github.com/JarvisChao/my-projects` in the address bar. The page displays instructions for setting up a repository. A red box highlights the copy link button next to the HTTPS URL `https://github.com/JarvisChao/my-projects.git`. Below the URL, there's a note about creating a new file or uploading an existing one, and a link to the GitHub repository settings.

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

Set up in Desktop or **HTTPS** `https://github.com/JarvisChao/my-projects.git` [Copy](#)

Get started by creating a new file or uploading an existing one. We recommend every repository include a `README`, `LICENSE`, and `.gitignore`.

複製到 Sourcetree 遠端設定裡

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

```
echo "# my-projects" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/JarvisChao/my-projects.git
git push -u origin main
```

...or push an existing repository from the command line

```
git remote add origin https://github.com/JarvisChao/my-projects.git
git branch -M main
git push -u origin main
```

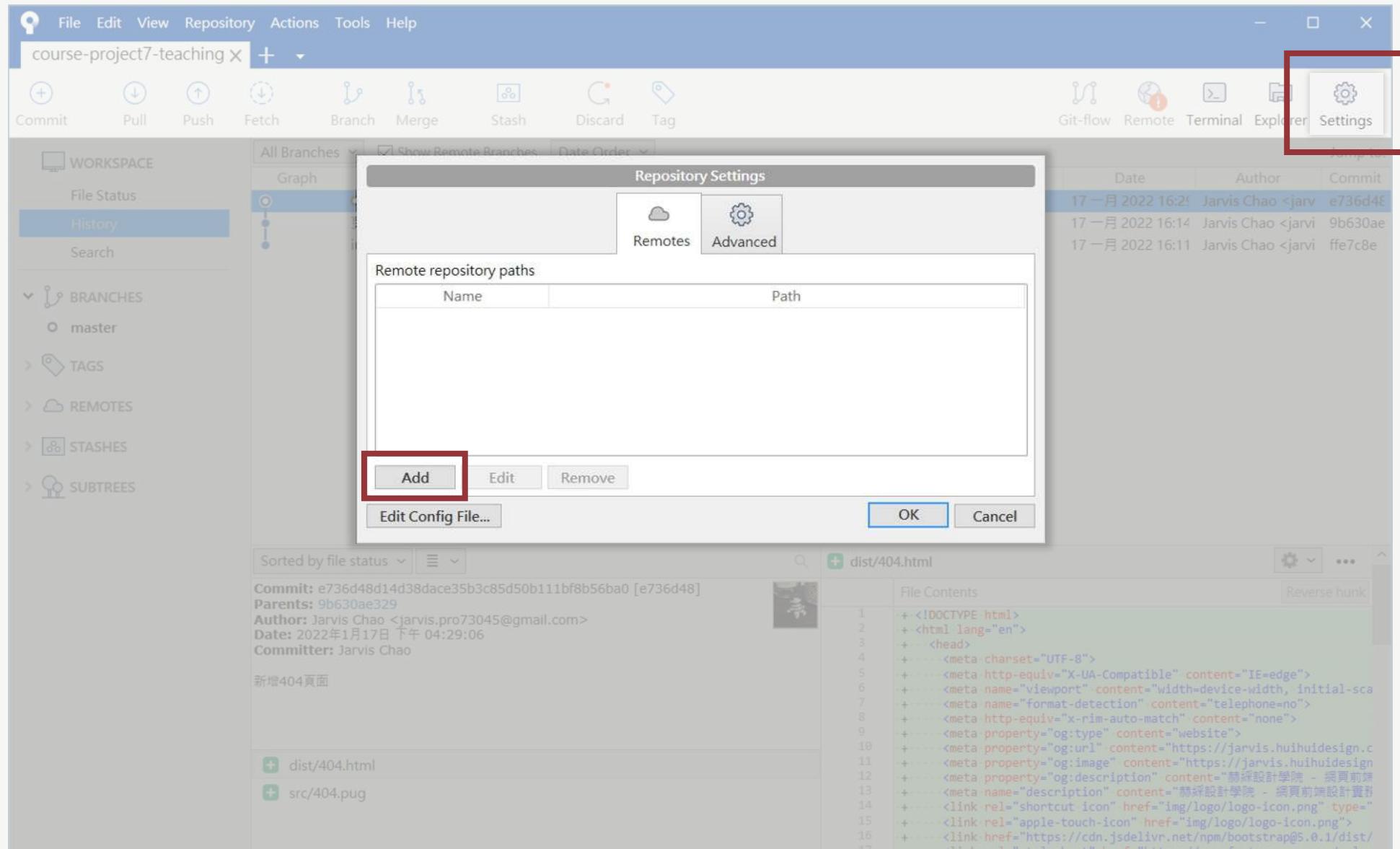
...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

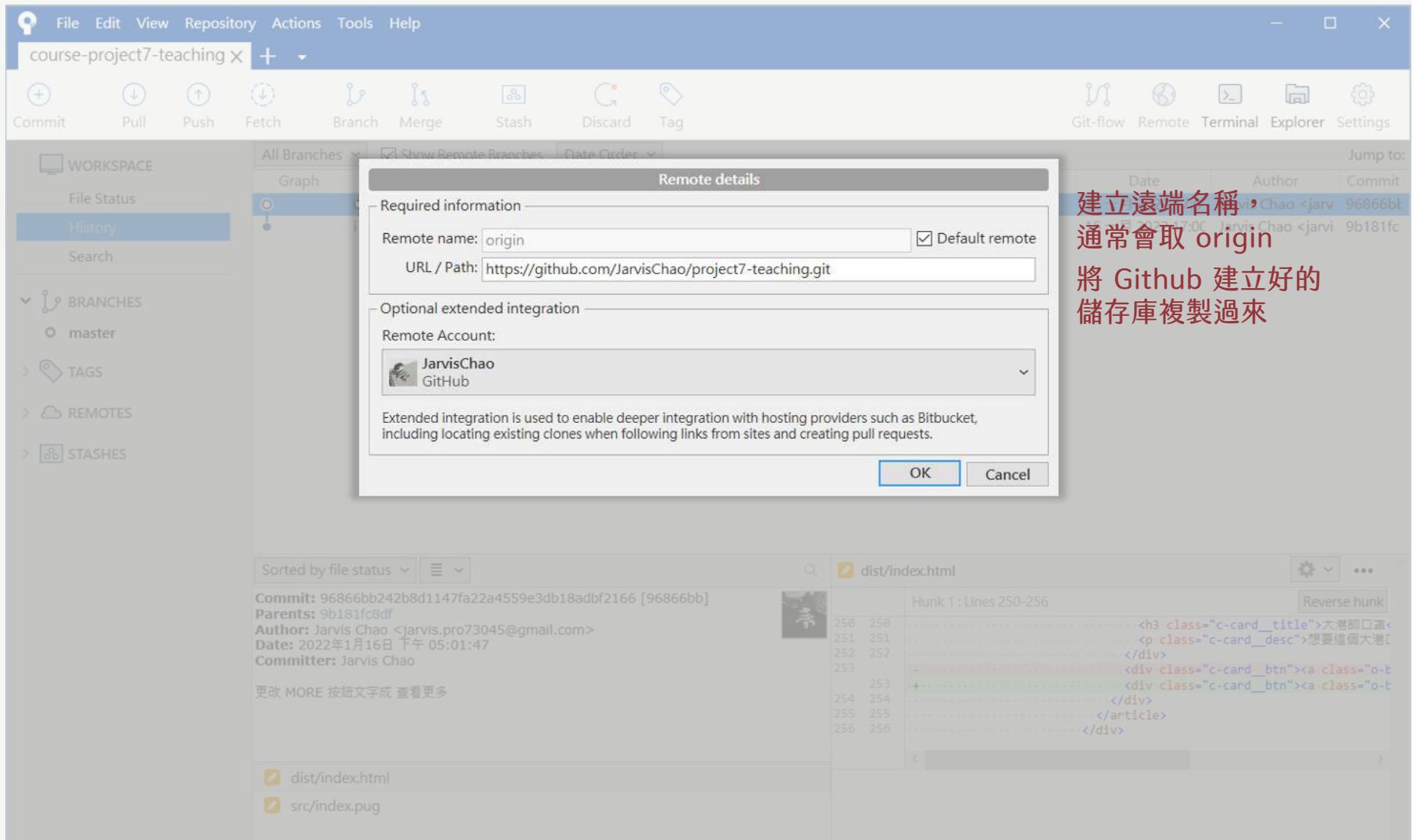
[Import code](#)

ProTip! Use the URL for this page when adding GitHub as a remote.

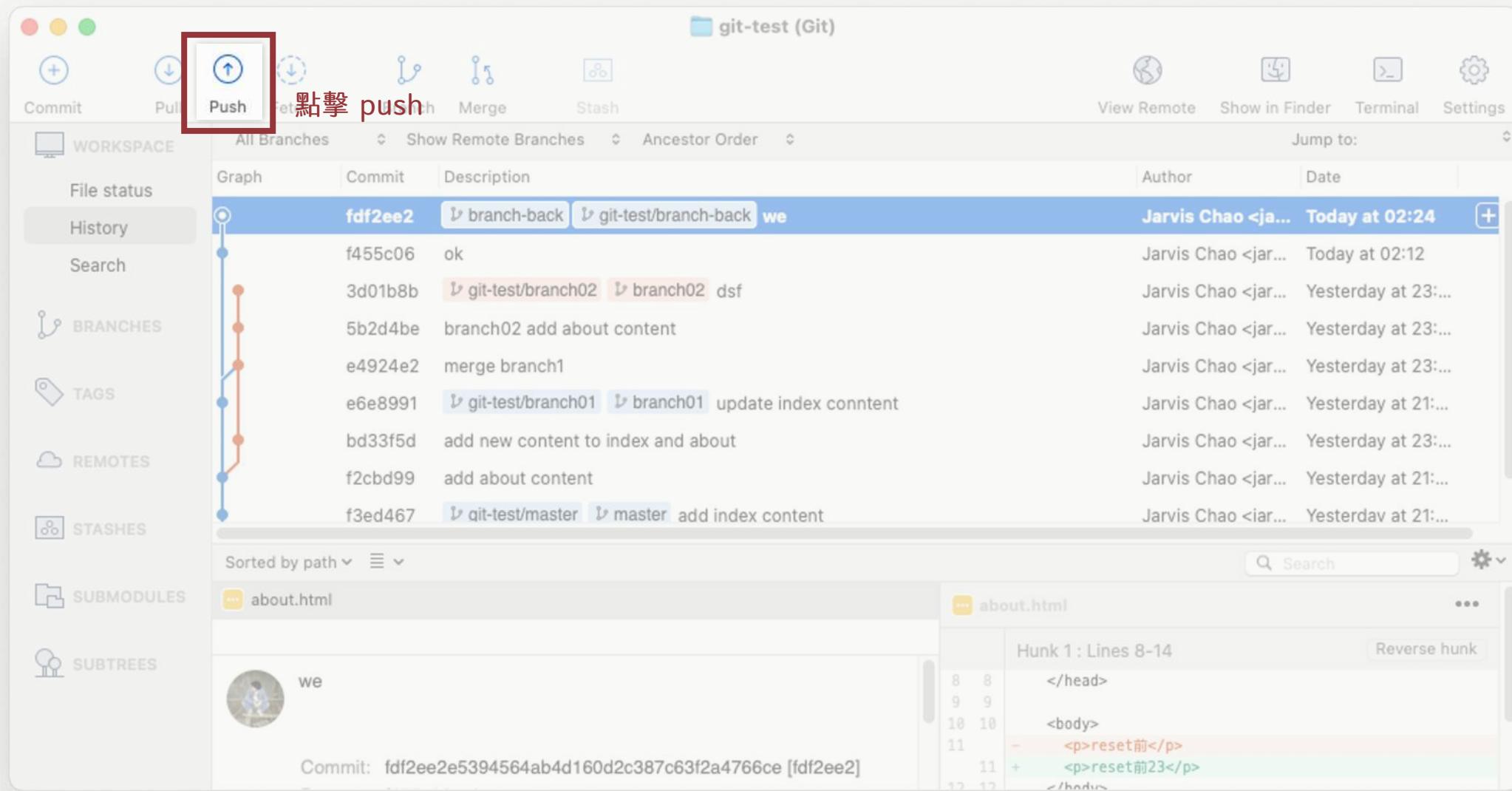
Sourcetree 連結遠端庫



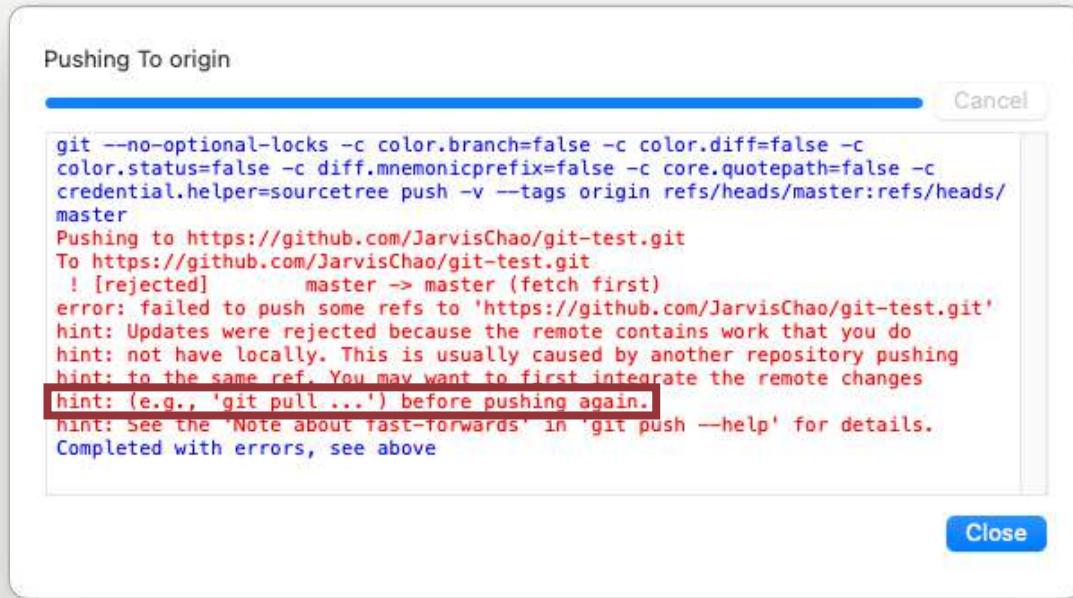
Sourcetree 連結遠端庫



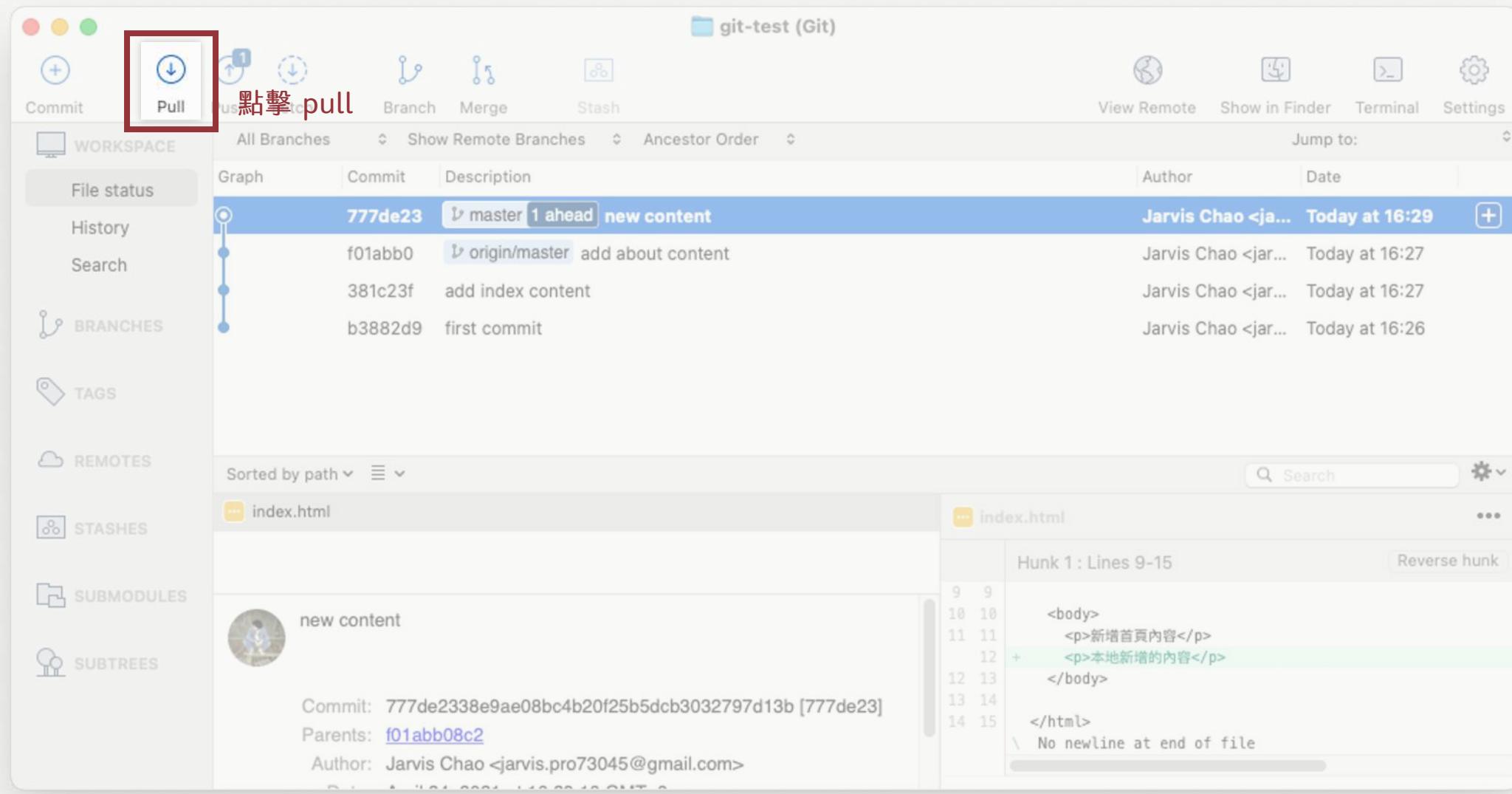
Sourcetree (push)



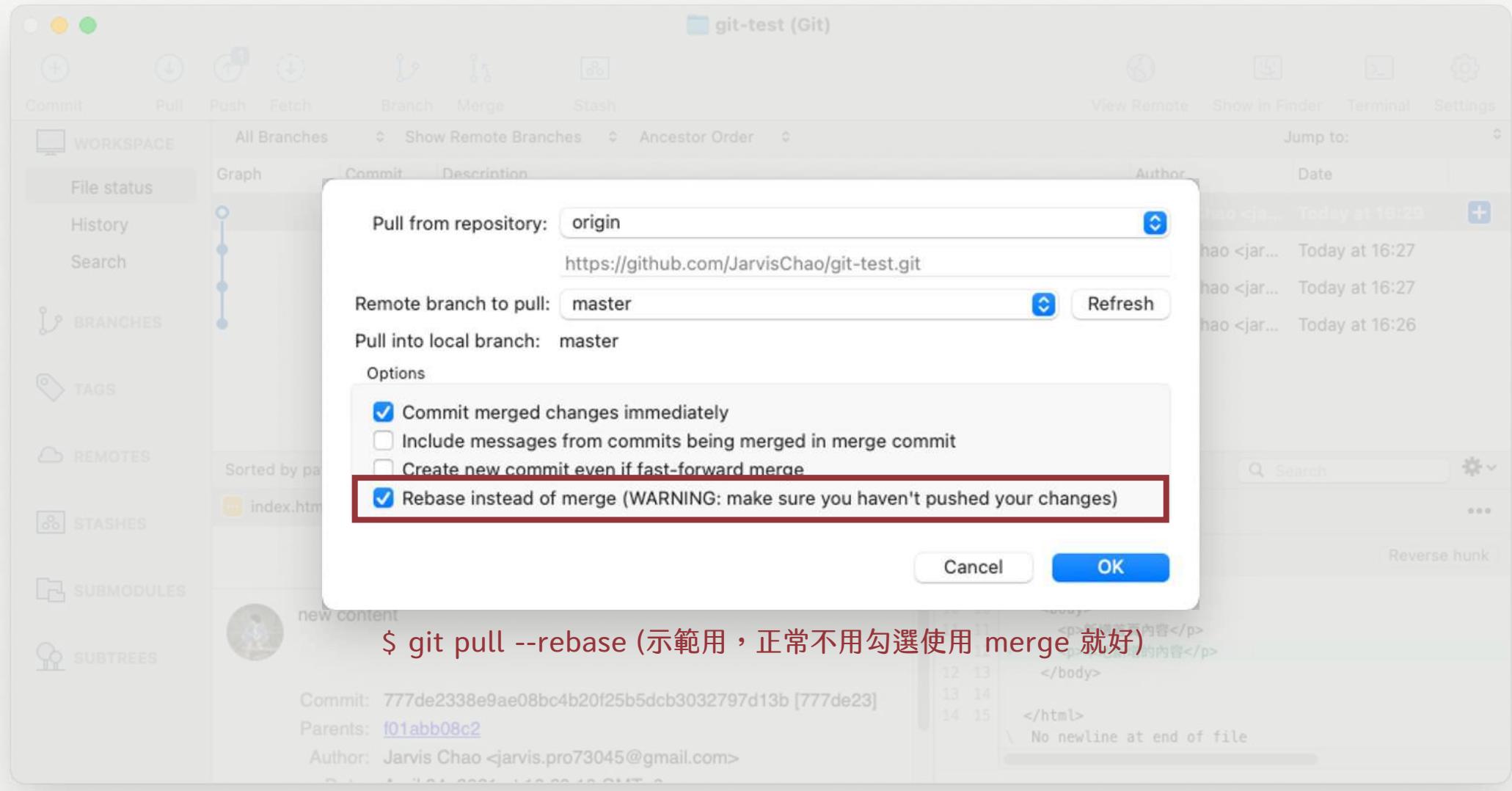
Push 時發生衝突



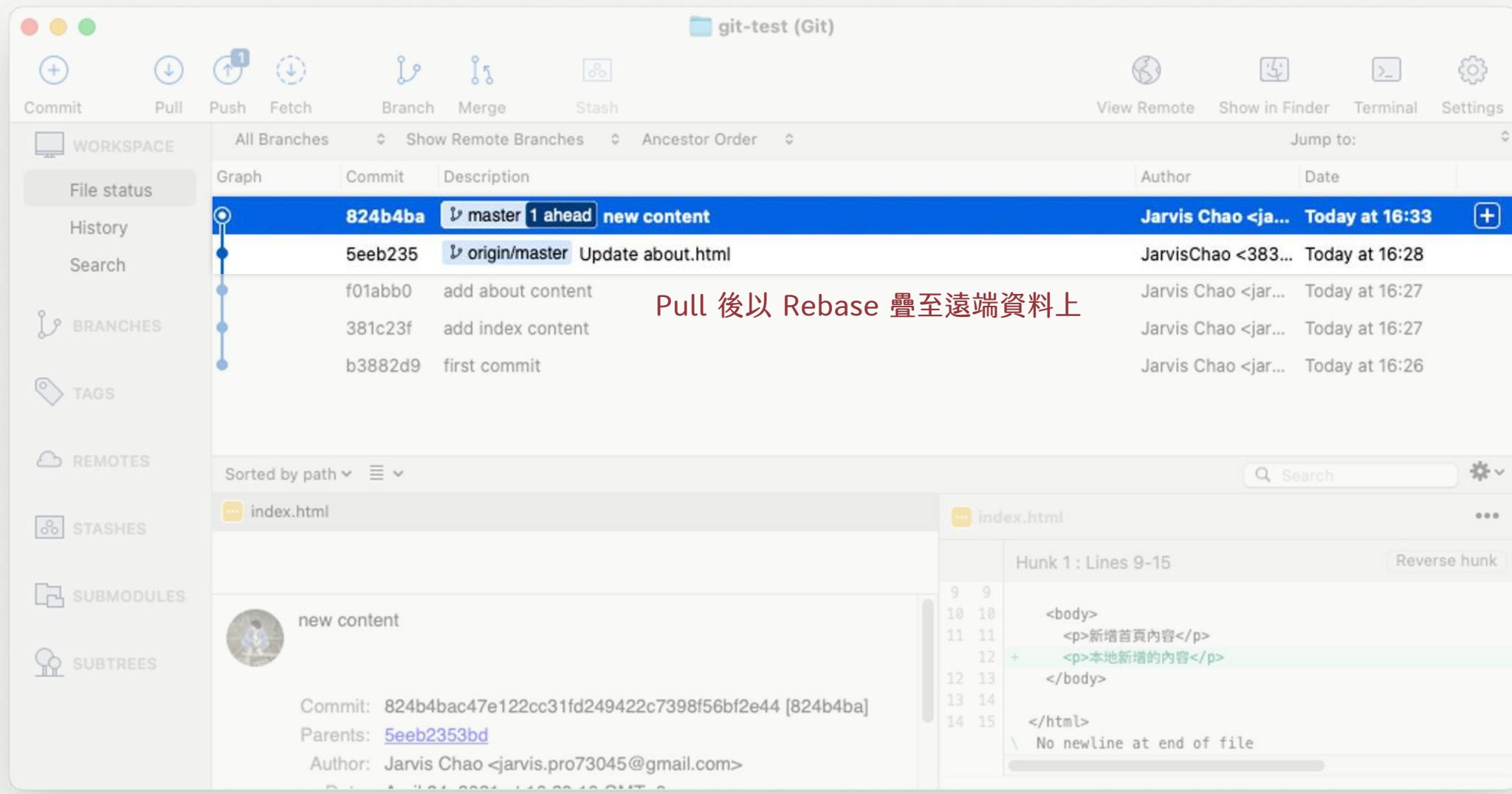
Sourcetree (pull)



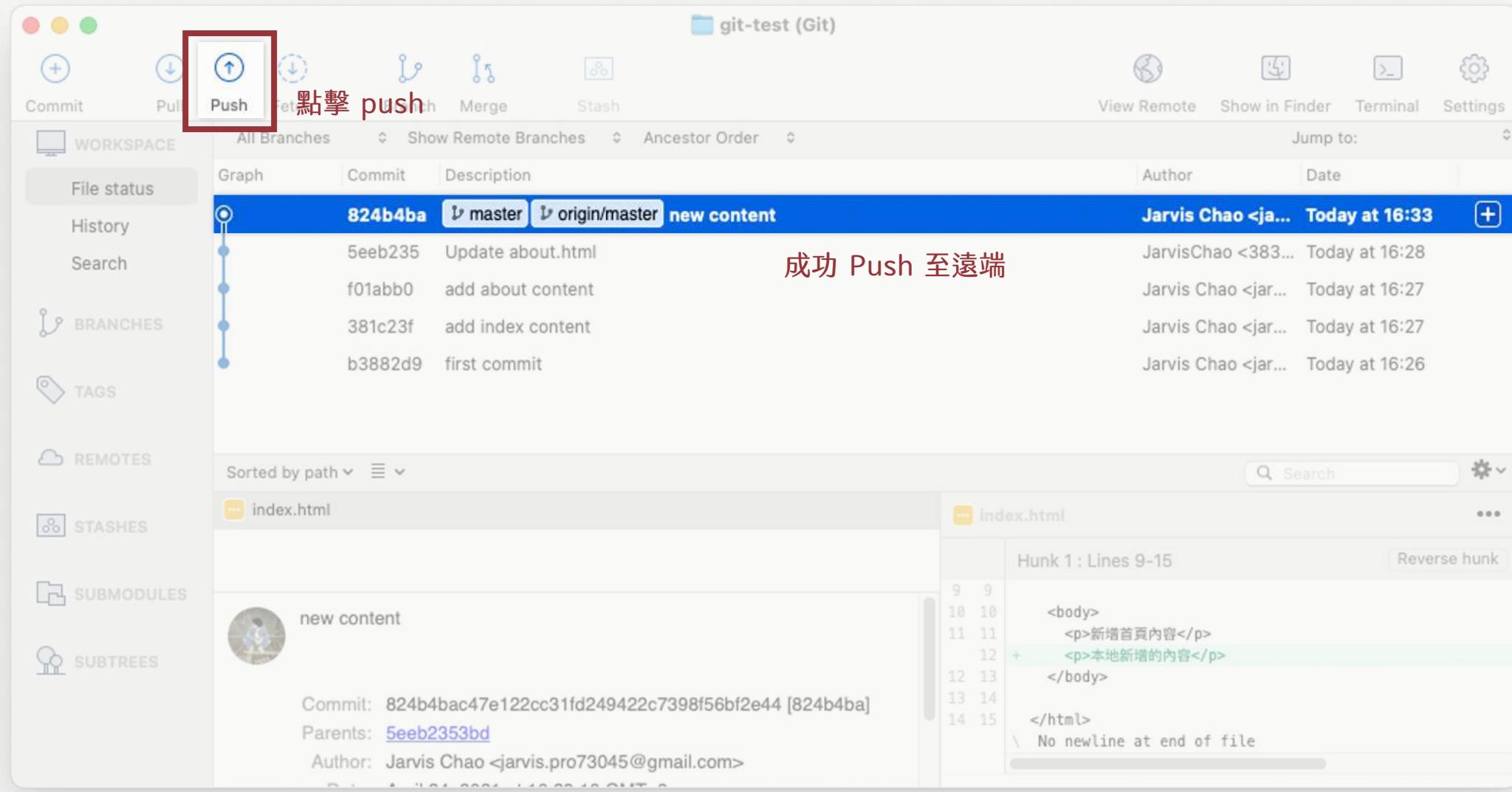
Sourcetree (pull)



Sourcetree (pull)



Sourcetree (push)



Sourcetree (無法 push ?)

如遇到無法成功 push，可以將 Remote 的 URL / Path 更改為以下形式：

`https://[USERNAME]:[ACCESS_TOKEN]@github.com/[USERNAME]/[REPO].git`

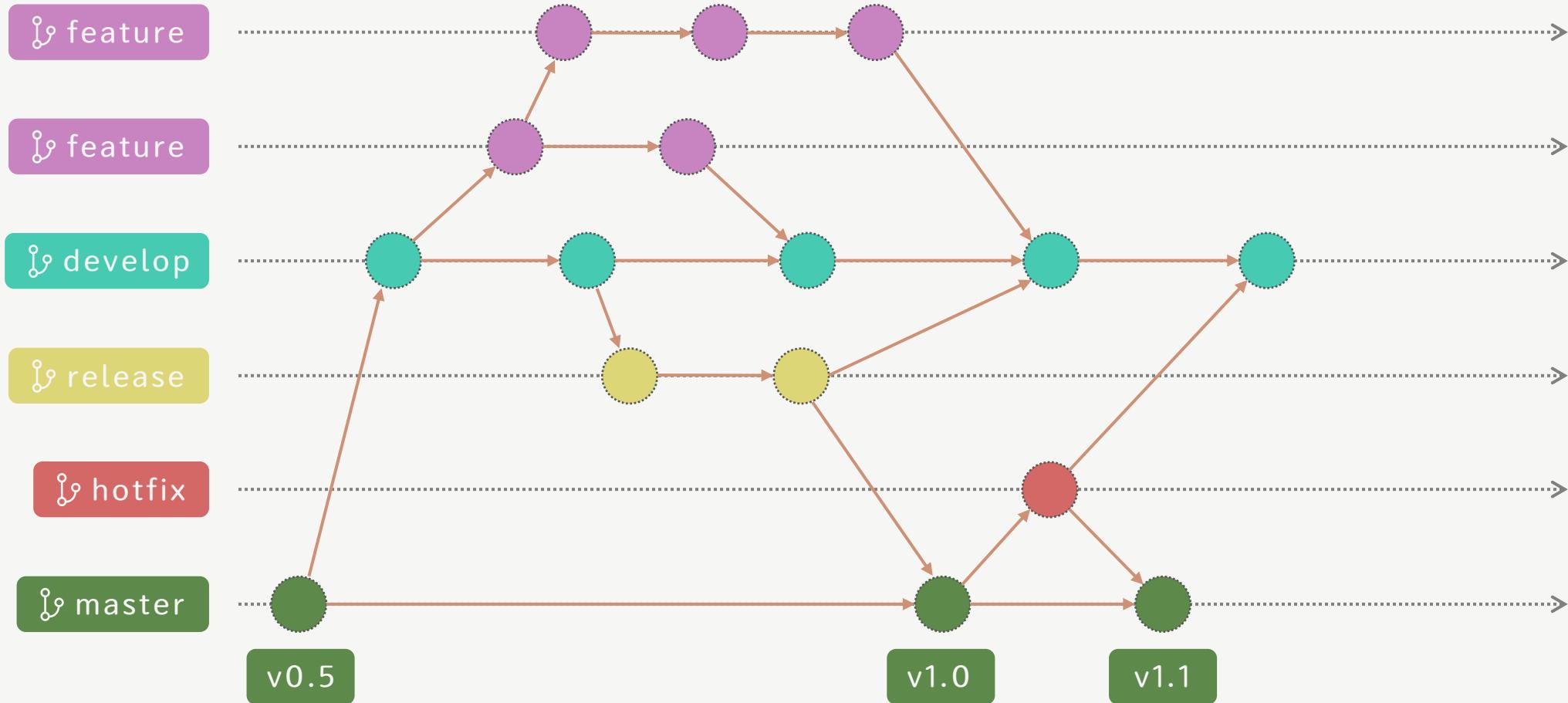
Ex. `https://JarvisChao:輸入取得的金鑰@github.com/JarvisChao/project7.git`

忽略檔案 .gitignore

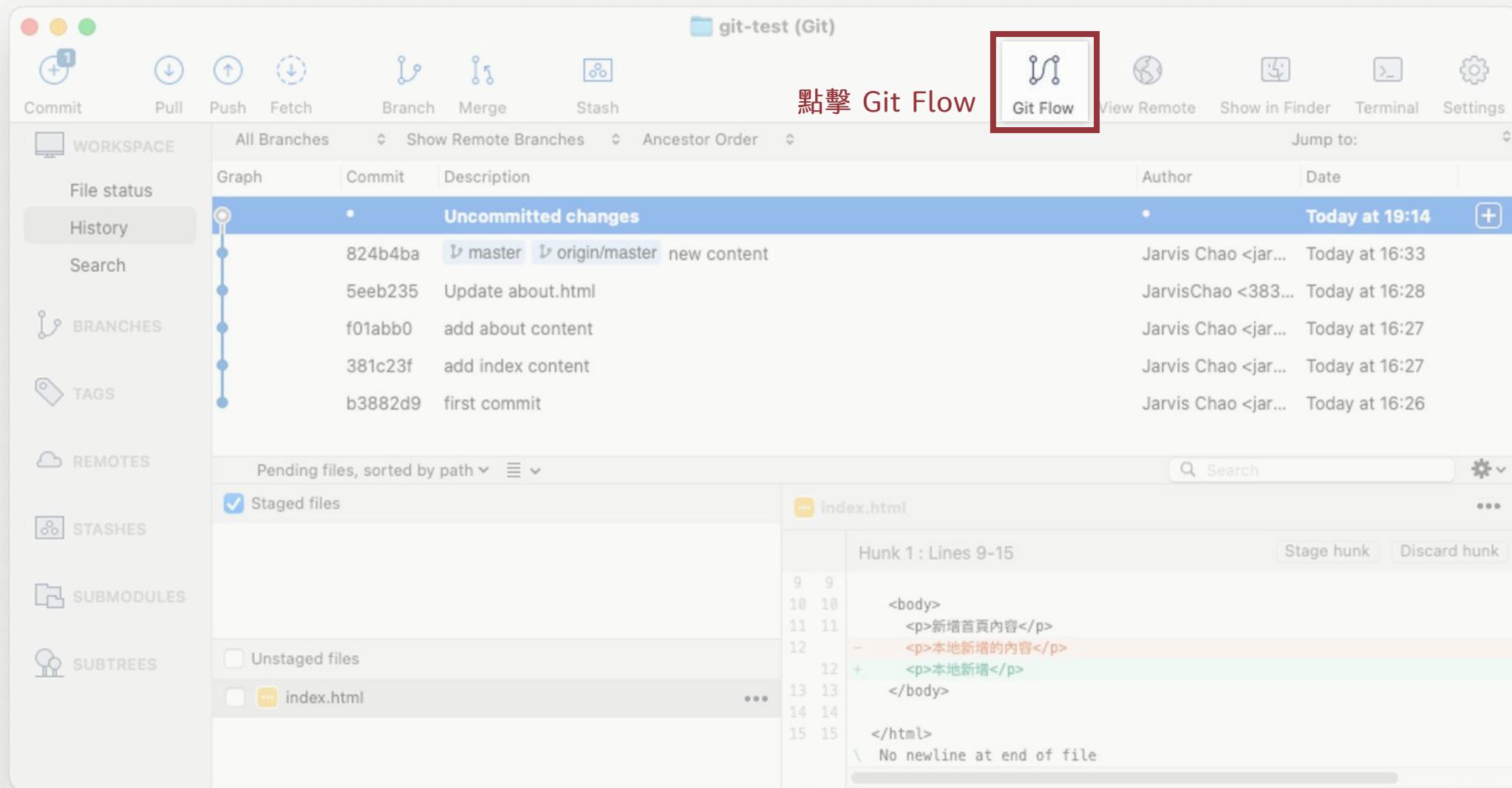
Git Flow

使開發更好管理的 Git 流程

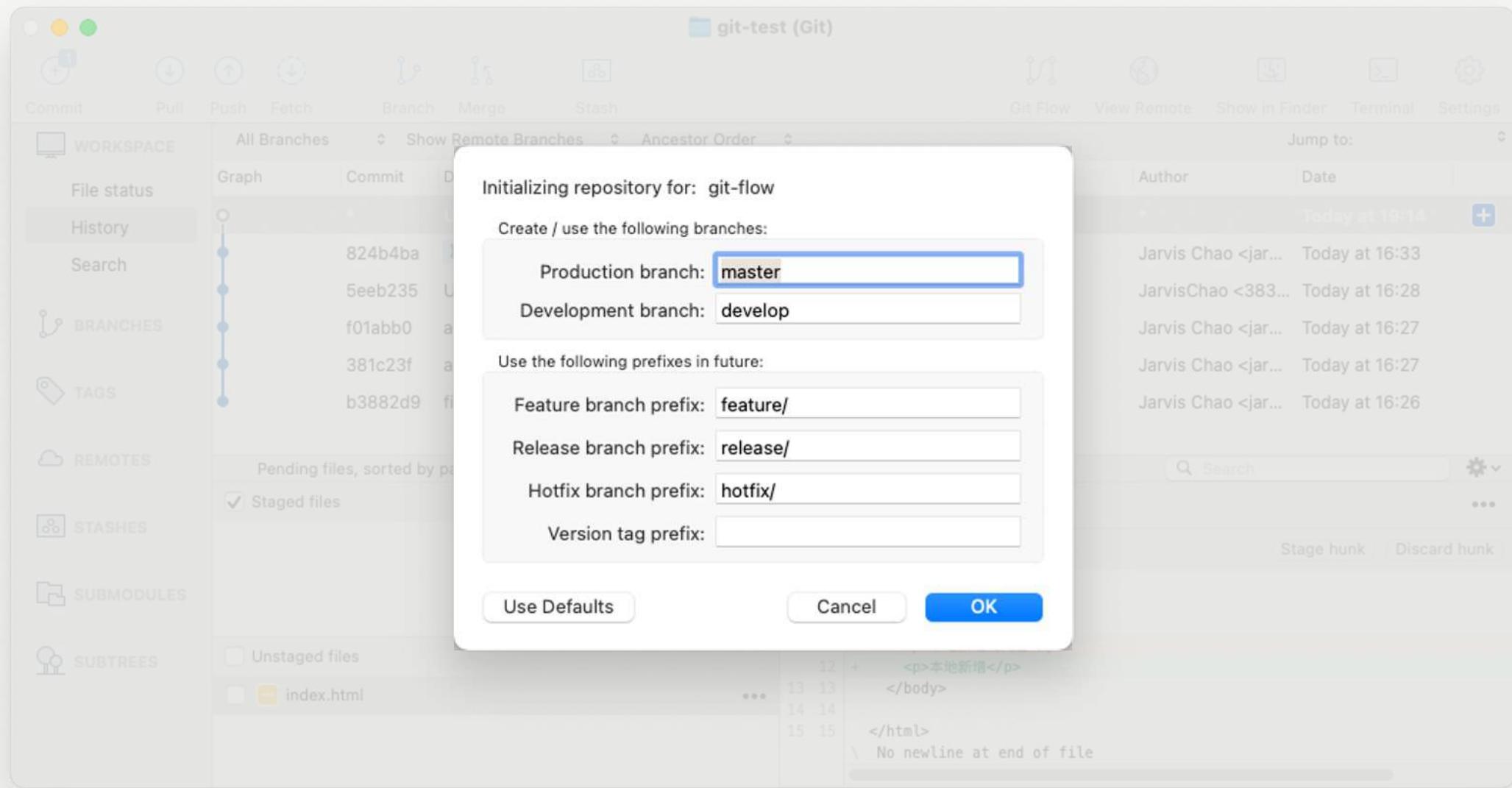
Git Flow



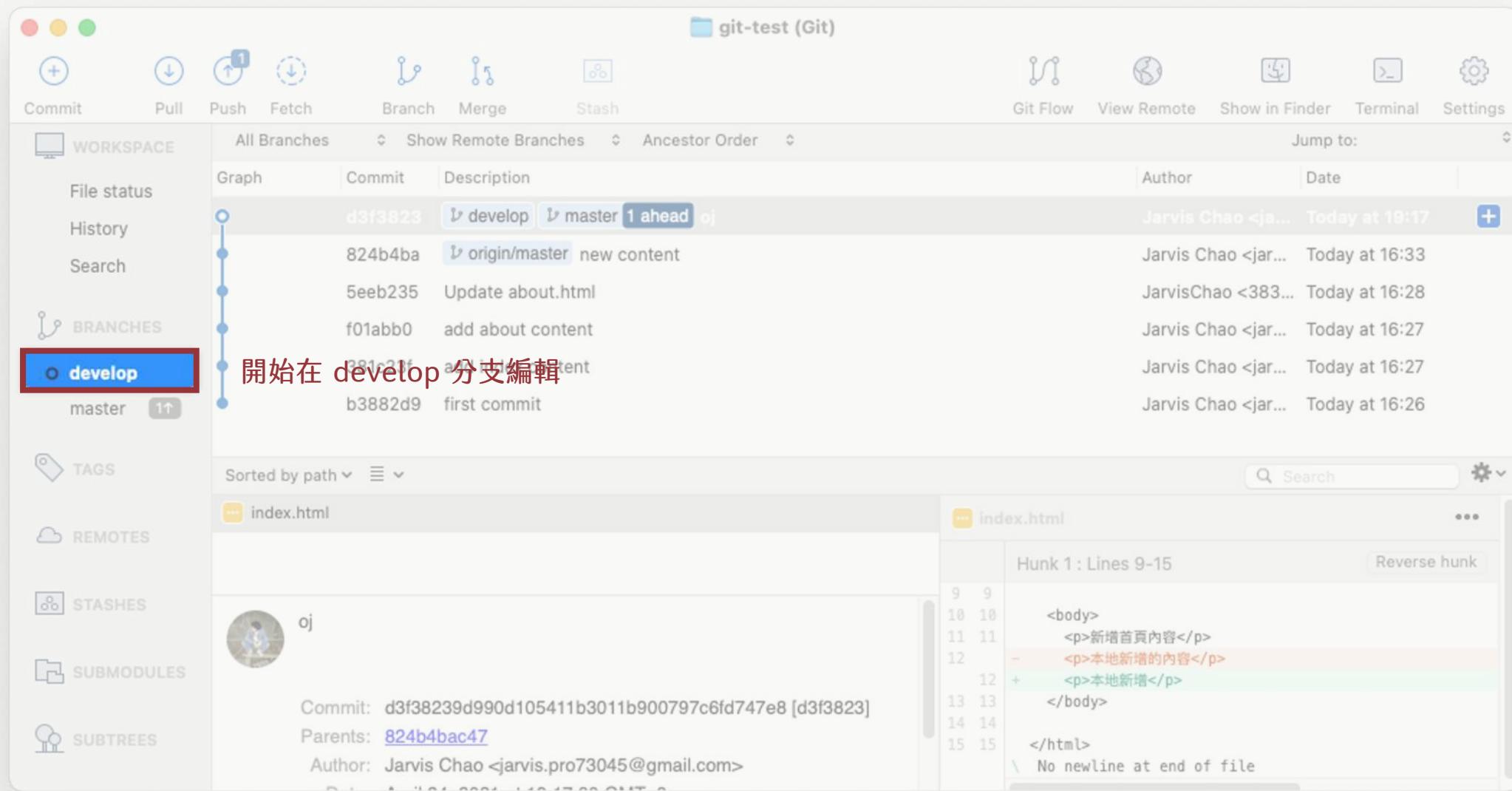
Sourcetree (git flow)



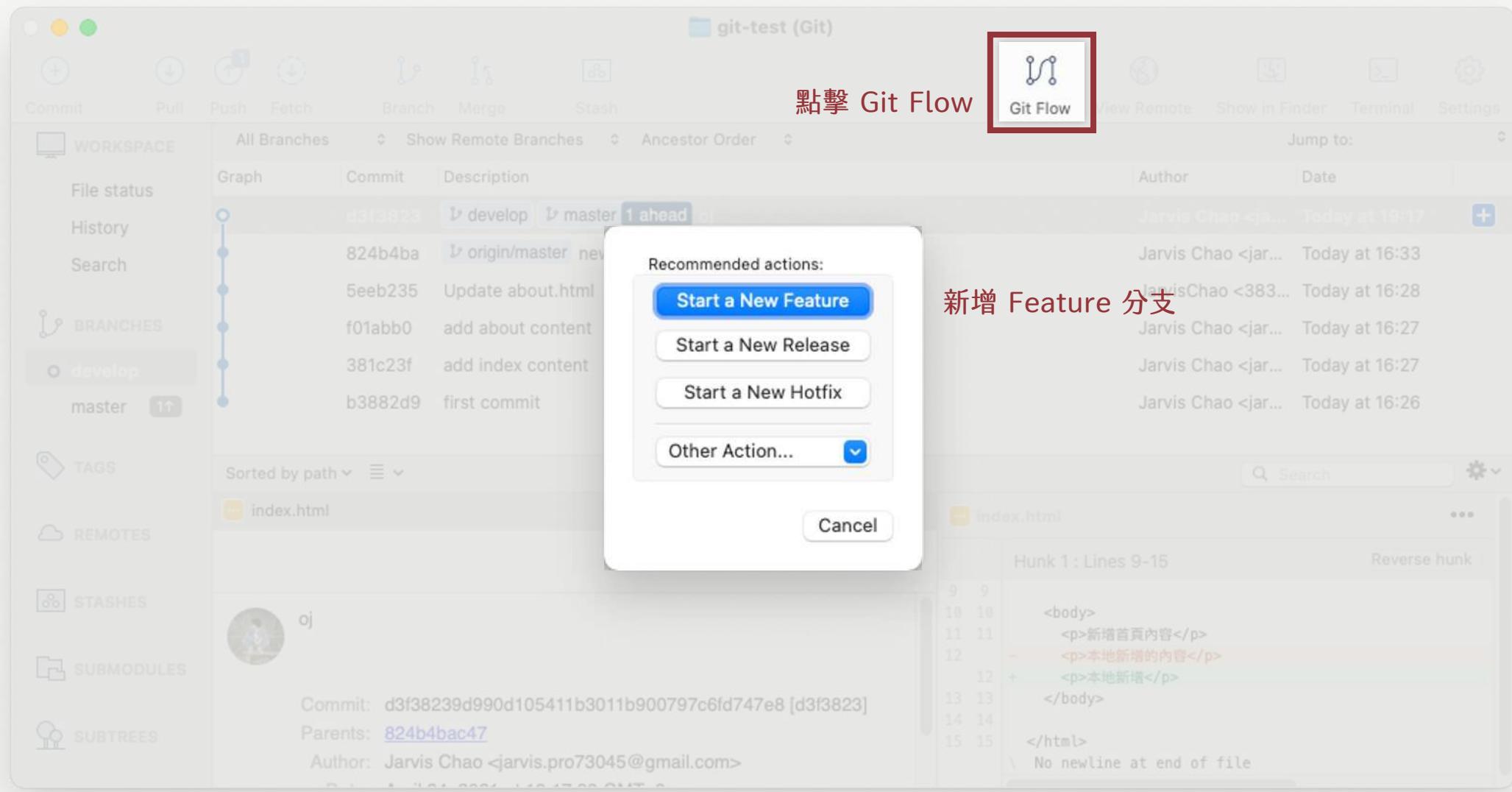
Sourcetree (git flow)



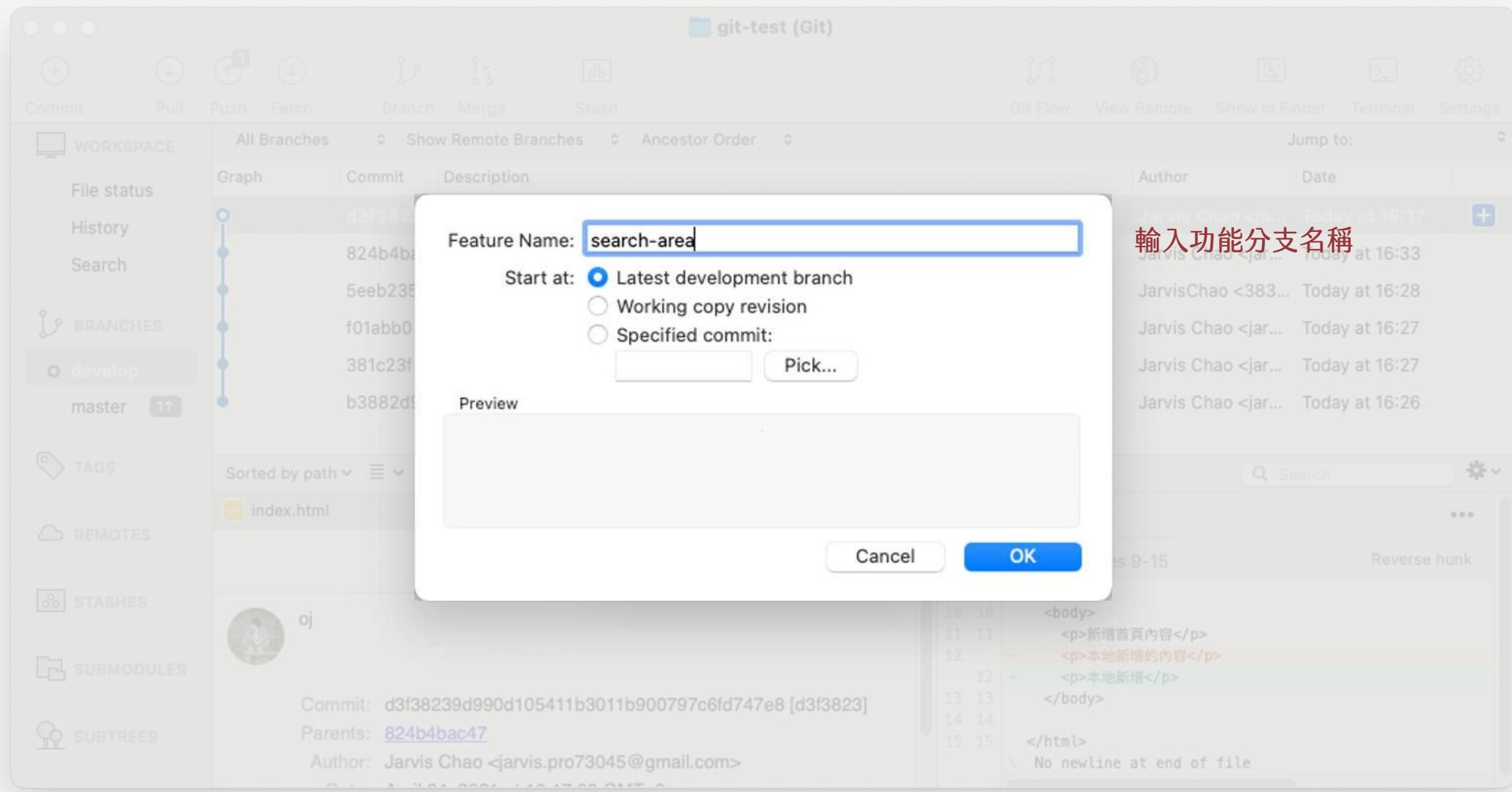
Sourcetree (git flow)



Sourcetree (git flow)



Sourcetree (git flow)



Sourcetree (git flow)

The screenshot shows the Sourcetree interface for a repository named "git-test (Git)". The main view displays a commit history for the "feature/search-area" branch. The commits are listed in chronological order from bottom to top:

- b3882d9 first commit
- 381c23f add index content
- f01abb0 add about content
- 5eeb235 Update about.html
- 824b4ba origin/master new content
- d3f3823 master 1 ahead develop
- 9465ab1 feature/search-area new search** (selected commit)

A red annotation text overlay on the right side of the commit list reads: "成功新增 feature/search-area 分支 並 commit 一次".

The bottom right pane shows the file contents of "search.html" with the following code:

```
+ <!DOCTYPE html>
+ <html lang="en">
+ <head>
+   <meta charset="UTF-8">
+   <meta http-equiv="X-UA-Compatible" content="IE=edge"
+   <meta name="viewport" content="width=device-width, i
1 <title>Document</title>
2 </head>
3 <body>
```