

[VSCode] 使用 Settings Sync 同步 VScode 的擴充功能與各項設定

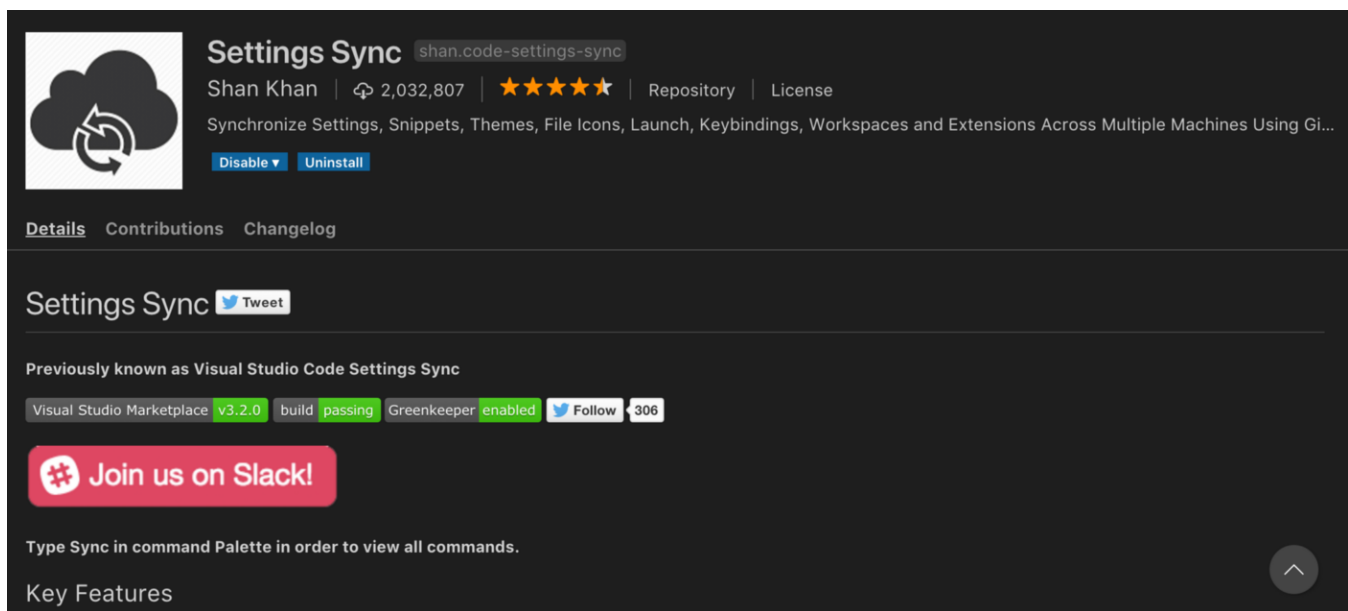
vscode 本身並沒有同步設定的功能，但是這個擴充功能可以達到相同的效果

由於本身工作上開發環境使用的是 VSCode (visual studio code)，並且有多台電腦使用 vscode 的需求。

一直以來，我都必須在多台電腦手工「同步」我修改的指令、新增的外掛…。這著實是一個很大的負擔。

最近得知在 VSCode Extension market 中有一款可以做到同步 VSCode 各項設定的外掛工具，也就是今天要來介紹的 — Settings Sync。

這個外掛其實是使用 [gist](#) 來同步你的設定，因此在使用這個同步工具時，請注意先辦好一個 github 帳號。



The image shows the GitHub repository page for 'Settings Sync' by Shan Khan. The repository has 2,032,807 stars and is a repository license. The description is 'Synchronize Settings, Snippets, Themes, File Icons, Launch, Keybindings, Workspaces and Extensions Across Multiple Machines Using Gi...'. There are buttons for 'Disable' and 'Uninstall'. Below the repository name, there are links for 'Details', 'Contributions', and 'Changelog'. A 'Settings Sync' tweet is shown. Below that, it says 'Previously known as Visual Studio Code Settings Sync'. There are badges for 'Visual Studio Marketplace', 'v3.2.0', 'build', 'passing', 'Greenkeeper', 'enabled', 'Follow', and '306'. A red button says 'Join us on Slack!'. Below that, it says 'Type Sync in command Palette in order to view all commands.' and 'Key Features'.

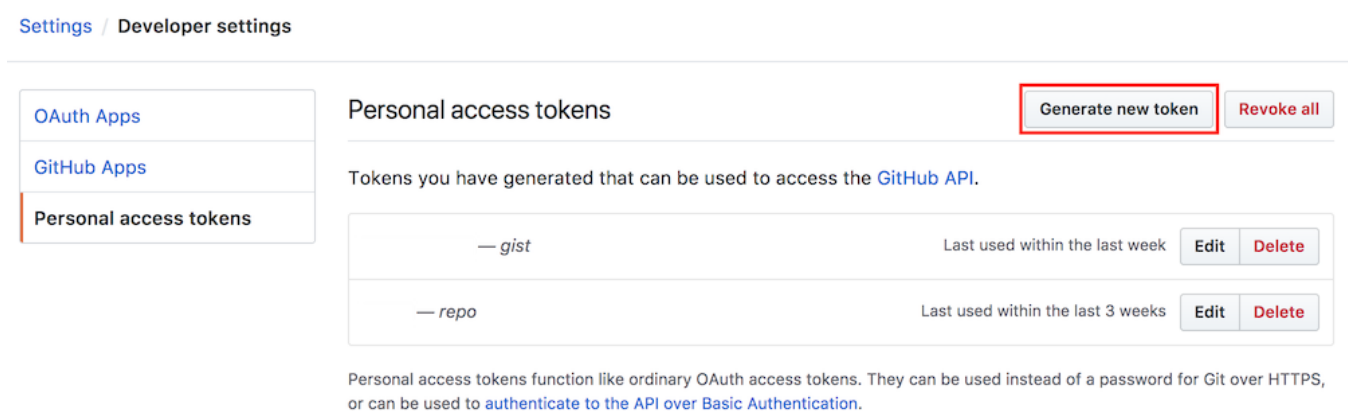
其實這個套件的作者，在如何上傳設定組態的部分已經寫得算是完整了，所以在這邊我會用翻譯的方式將作者的使用教學重現一遍。

上傳(Upload/Update Setting)

這個外掛需要使用你在 GitHub 上的 Access Token。你可以從依照以下的步驟很簡單的創造一個新的 Token（注意：創造的 Token 要有 Gist 的權限）。

首先到 [Settings](#) / [Developer settings](#) / [Personal access tokens](#) /

然後選擇 *Generate New Token*



The image shows the GitHub Developer settings page for 'Personal access tokens'. The left sidebar has links for 'OAuth Apps', 'GitHub Apps', and 'Personal access tokens'. The main content area has a 'Generate new token' button (highlighted with a red box) and a 'Revoke all' button. Below these buttons, it says 'Tokens you have generated that can be used to access the GitHub API.' There is a table with two rows: one for '— gist' and one for '— repo'. Each row has a 'Last used' column and 'Edit' and 'Delete' buttons. The 'Last used' column for 'gist' says 'Last used within the last week' and for 'repo' says 'Last used within the last 3 weeks'. Below the table, it says 'Personal access tokens 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.'

<https://marketplace.visualstudio.com/items?itemName=Shan.code-settings-sync>

在權限範圍中勾選 Gist，然後點選 Generate token

[OAuth Apps](#)[GitHub Apps](#)**Personal access tokens**

New personal access token

Personal access tokens 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](#).

Token description

What's this token for?

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes](#).

<input type="checkbox"/> repo	Full control of private repositories
<input type="checkbox"/> repo:status	Access commit status
<input type="checkbox"/> repo_deployment	Access deployment status
<input type="checkbox"/> public_repo	Access public repositories
<input type="checkbox"/> repo:invite	Access repository invitations
<input type="checkbox"/> admin:org	Full control of orgs and teams
<input type="checkbox"/> write:org	Read and write org and team membership
<input type="checkbox"/> read:org	Read org and team membership
<input type="checkbox"/> admin:public_key	Full control of user public keys
<input type="checkbox"/> write:public_key	Write user public keys
<input type="checkbox"/> read:public_key	Read user public keys
<input type="checkbox"/> admin:repo_hook	Full control of repository hooks
<input type="checkbox"/> write:repo_hook	Write repository hooks
<input type="checkbox"/> read:repo_hook	Read repository hooks
<input type="checkbox"/> admin:org_hook	Full control of organization hooks
<input checked="" type="checkbox"/> gist	Create gists
<input type="checkbox"/> notifications	Access notifications
<input type="checkbox"/> user	Update all user data
<input type="checkbox"/> read:user	Read all user profile data
<input type="checkbox"/> user:email	Access user email addresses (read-only)
<input type="checkbox"/> user:follow	Follow and unfollow users
<input type="checkbox"/> delete_repo	Delete repositories
<input type="checkbox"/> admin:pgp_key	Full control of user pgp keys (Developer Preview)
<input type="checkbox"/> write:pgp_key	Write user pgp keys
<input type="checkbox"/> read:pgp_key	Read user pgp keys

Generate token[Cancel](#)

<https://marketplace.visualstudio.com/items?itemName=Shan.code-settings-sync>

將產生出來的 Token 好好保存（注意！這個 Token 絕對不能不見，因為下次進來是再也看不到的喔！）。

接下來就能夠利用剛剛的 Token 來同步 VSCode 的設定啦！

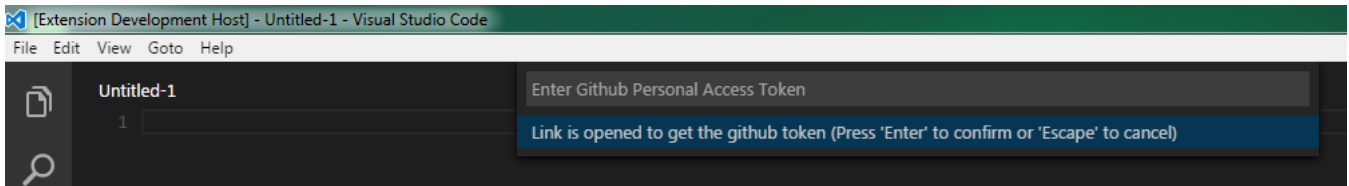
輸入 Shift + Alt + U 熱鍵或是在 VScode 的 *Command Palette* 輸入

> Sync Update / upload Settings

(p.s.使用 $\text{⌘}+P$ 可以叫出 VSCode 的 Command Palette)

如此 Settings Sync 將會跳出一個小小的對話框，要求你貼上你專屬的使用者權限 Token(就是你剛剛存起來的那串 Token)。

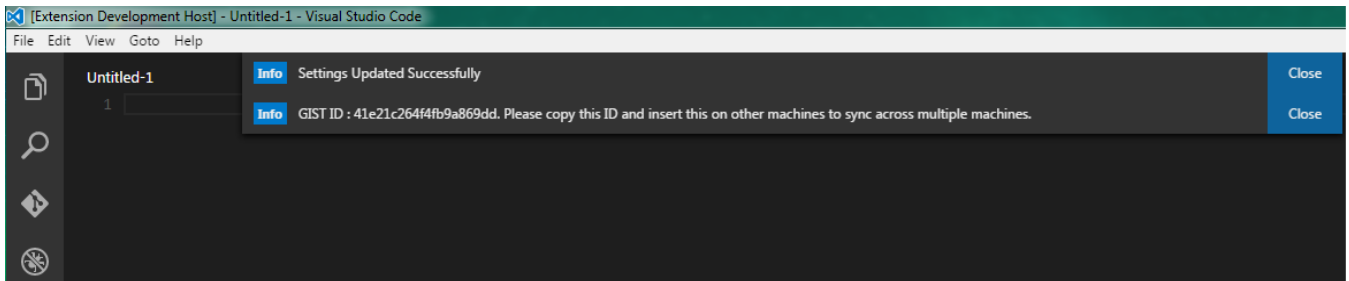
貼好剛剛保存的 Token 後，按下 Enter。



<https://marketplace.visualstudio.com/items?itemName=Shan.code-settings-sync>

Settings Sync 會自動上傳你的各種組態設定到你的 Gist。然後外掛會給你一組產生好的 **Gist ID**、**GitHub Token**、**Git Type**。

請記錄下來，Gist ID 將是用來下載你的組態所需的資訊。



<https://marketplace.visualstudio.com/items?itemName=Shan.code-settings-sync>

CODE SETTINGS SYNC UPLOAD SUMMARY

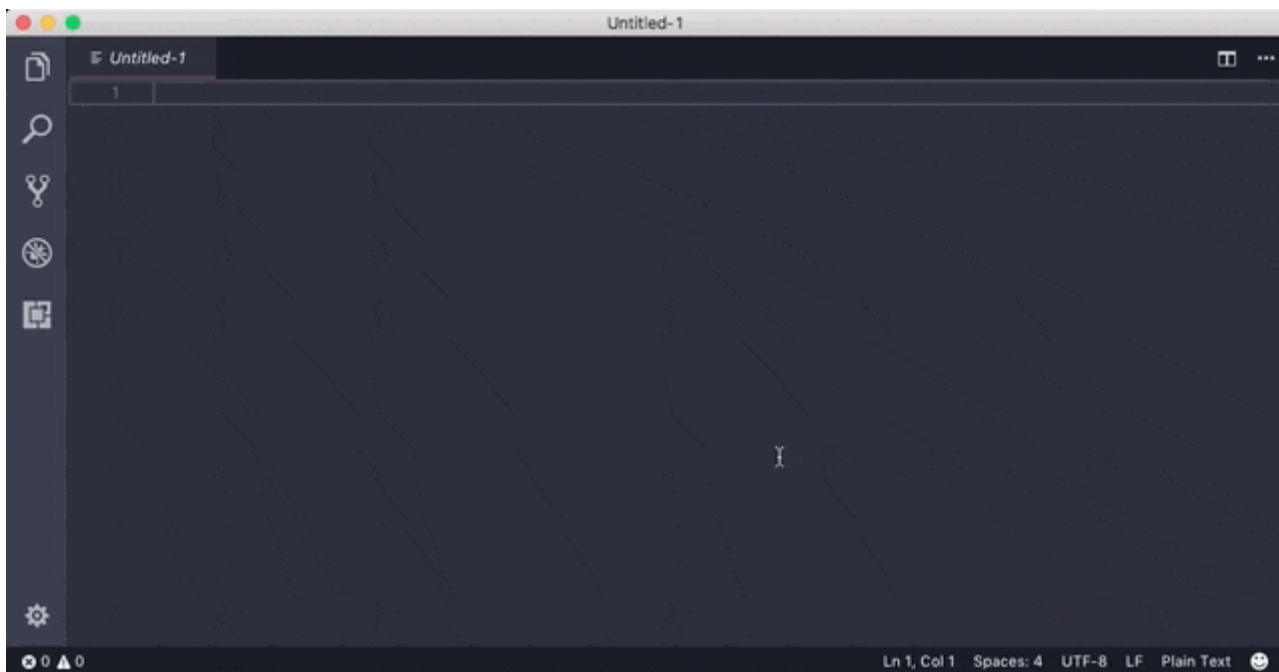
Version: x.x.x

GitHub Token: xxx

GitHub Gist: xoxoxoxoxoxoxoxoxoxoxoxoxoxoxoxoxoxo

GitHub Gist Type: Secret

以下是作者附上的一個完整 gif ，展示怎麼開始上傳。



<https://marketplace.visualstudio.com/items?itemName=Shan.code-settings-sync>

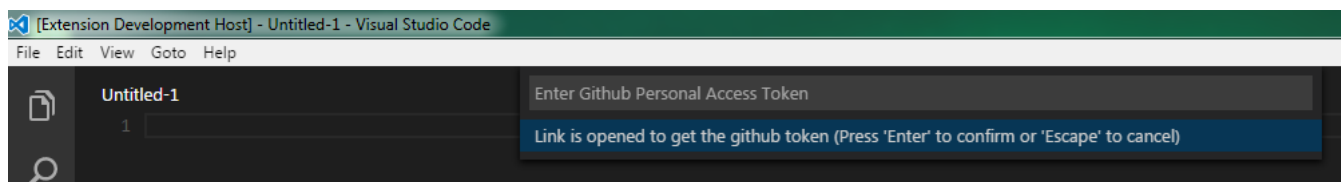
下載 (Download Settings)

使用 `Shift + Alt + D` 熱鍵或是在 VSCode 的 *Command Palette* 輸入

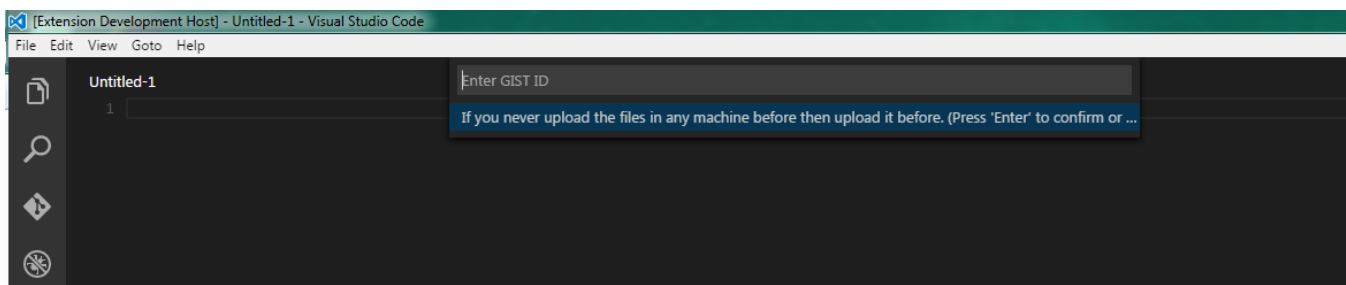
> *Sync Download Settings*

會跳出對話框要求你輸入你的 github Token。

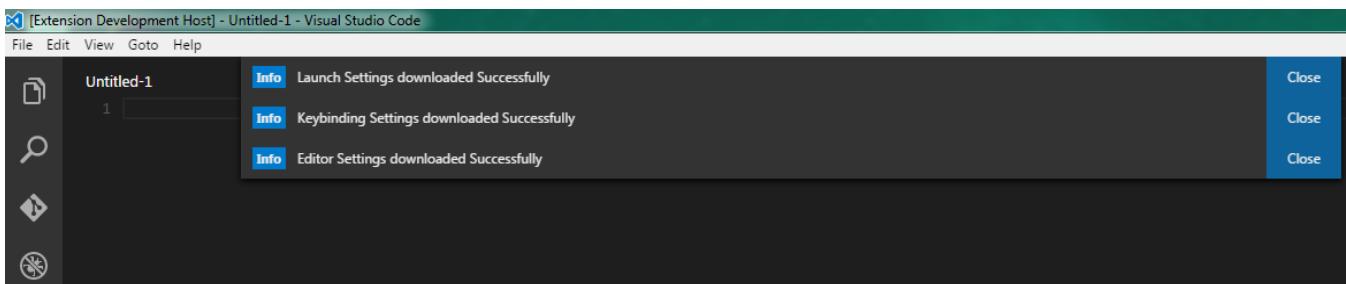
將你的 Token 輸入後，按下 Enter。



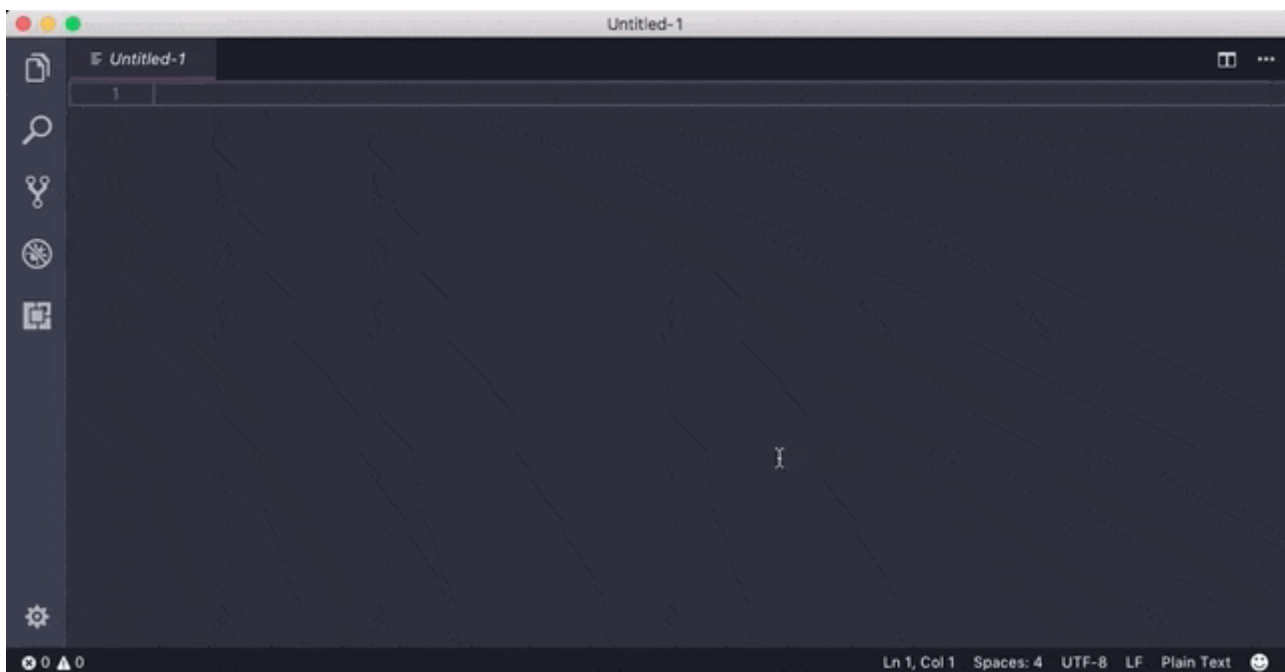
接下來外掛會要求你輸入你的 Gist Token



當你下載成功後，會跳出成功的訊息。



一樣，作者附上了完整的操作流程。



使用下載的例外狀況

若你要下載同步組態的 VSCode 本身已經有本地的外掛與設定。你可能會發現，當你在下載同步的組態時，依照上方的下載步驟輸入好 github Token 後，**Settings Sync** 會自動幫你創好一組新的 Gist Token。然而不會跳出要求輸入 Gist Token 的對話框。

這是因為，*Settings Sync* 認為你要同步本地的組態到 Gist。

這時若你還是執意要下載遠端的組態，你可以進到 VSCode 的 Settings。

使用

Perferences / Settings

在搜尋欄輸入關鍵字 `sync` 找到 *Settings Sync* 相關的設定。

在當中找到 `sync.gist` 的設定。

```
// Ask gist name upon creating. Helps you to identify the gist if you have multiple gists.
"sync.askGistName": false,

// Set it true to Auto Download the settings on code start. [Code Restart Required]
"sync.autoDownload": false,

// Set it true to Auto Upload on the settings change. [Code Restart Required]
"sync.autoUpload": false,

// Set it to true if you want to download the settings even when you have the latest settings.
"sync.forceDownload": false,

// GitHub GIST ID for Settings Sync.
"sync.gist": "",

// When set to true, will show the result in status bar instead of summary page.
"sync.quietSync": false,

// Set it to false if you dont want to remove extensions while downloading.
"sync.removeExtensions": true,

// Set it to false if you dont want to upload / download the extensions.
"sync.syncExtensions": true
```

在這邊將你的 gist ID 貼過去，之後使用同步的熱鍵 `Shift + Alt + D` 或是使用 VSCode 的 *Command Palette* 輸入

> *Sync Download Settings*

之後你就會發現成功下載了。

但是這個做法有一個很大的風險，此作法會覆蓋掉所有原本的外掛與設定。請謹慎使用。

後記

VSCode 雖然不是地表最強 IDE。但是他在我心中也算是地表最強編輯器了。很多好用的擴充套件、方便的熱鍵手勢，在在地加速程式上的開發速度。

身為一個重度 VSCode 使用者，同步功能實在是太重要了。因此做了這篇記錄，希望能夠幫助到來閱讀的每一個人。

以上就是使用 *Settings Sync* 的使用方式了。目前我把我所碰到的坑，與解法都寫在了這篇。