# ETF交接文件

目錄

[ETF交接文件 1](#_Toc179894144)

[帳號權限 2](#_Toc179894145)

[創建帳號 2](#_Toc179894146)

[登入 4](#_Toc179894147)

[登出 4](#_Toc179894148)

[檢視ETF列表（公告中、募資中）[未登入] 6](#_Toc179894149)

[頁面切換 Pagination 7](#_Toc179894150)

[Elasticsearch搜尋功能（前端、後端） 7](#_Toc179894151)

[名稱&單號搜尋 8](#_Toc179894152)

[類別搜尋 8](#_Toc179894153)

[時長搜尋 9](#_Toc179894154)

[時間區段搜尋 9](#_Toc179894155)

[管理ETF（創建、加入、其他…等等）[登入] 10](#_Toc179894156)

[創建ETF 11](#_Toc179894157)

[加入狀況 13](#_Toc179894158)

[創建（Created） 13](#_Toc179894159)

[加入（Joined） 13](#_Toc179894160)

[其他（Other） 14](#_Toc179894161)

[時間狀態 14](#_Toc179894162)

[未來式（Future） 14](#_Toc179894163)

[公告中（Announcing） 14](#_Toc179894164)

[募資中（Fundraising） 15](#_Toc179894165)

[進行中（Progressing） 15](#_Toc179894166)

[已結束（Completed） 15](#_Toc179894167)

[已關閉（Closed） 16](#_Toc179894168)

TODO

* 1. 退出ETF

1. ETF詳細頁面[登入]
   1. 加入ETF功能（不可加入自己創建的ETF）
   2. 刪除ETF功能（自己創建的ETF，在沒有人加入的狀態下可刪除）

### 開啟服務

#### Elasticsearch搜尋引擎

執行文件（Documents）內的「elasticsearch.bat - 捷徑」或開啟E:\Program Files\elasticsearch-8.15.1\bin\elasticsearch.bat即可。開啟後需要花一段時間運行才能正常運作，建議提早開啟避免等待。

#### PostgreSQL資料庫

執行文件（Documents）內的「postgre\_run.bat」即可。也可以開啟小黑窗或VS code下方命令列，輸入pg\_ctl start -D "E:\Program Files\PostgreSQL\16\data"。後端需要接到它才有辦法處理資料，建議在開啟後端之前先開啟。

#### Django後端

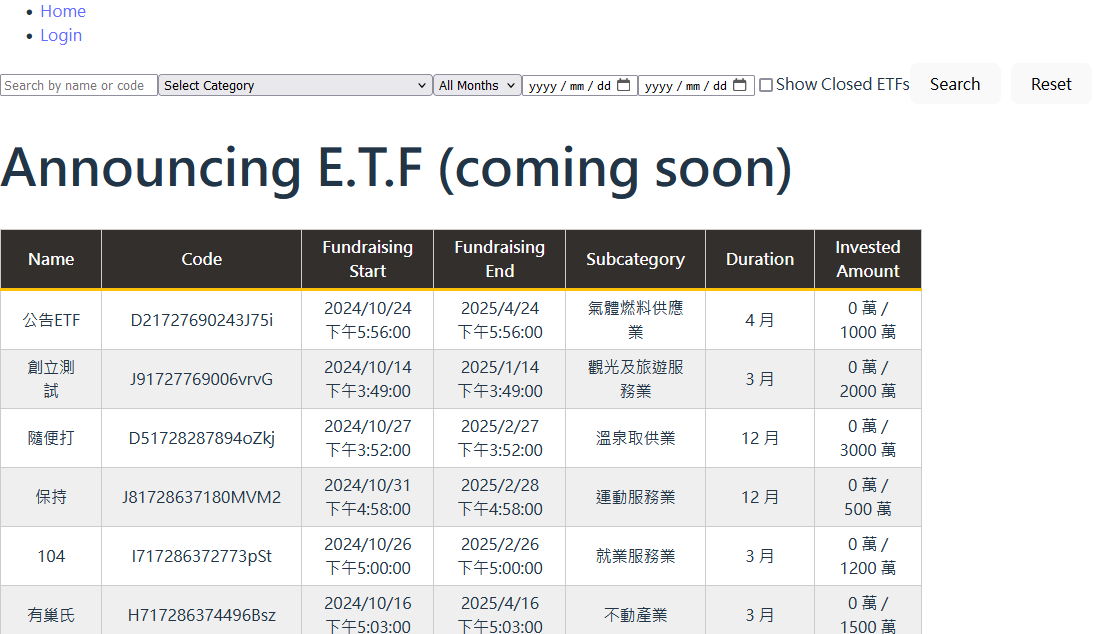
開啟小黑窗或VS code下方命令列，切換到後端資料夾（backend）下，執行python manage.py runserver。

#### React前端

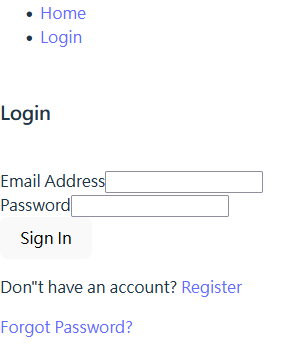
開啟小黑窗或VS code下方命令列，切換到前端資料夾（frontend）下，執行yarn dev。

### 帳號權限

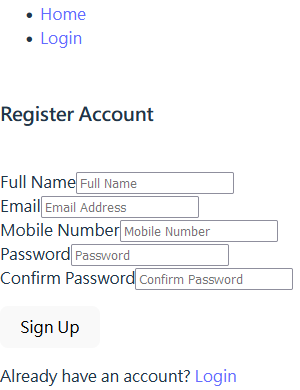
#### 創建帳號



首先，從上方的Login進入登入頁面。



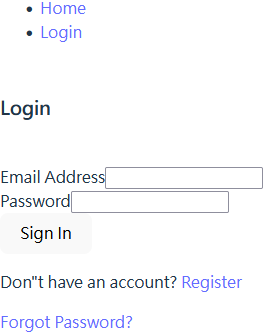
點擊下方的Register後，進入註冊畫面（如下圖）：



輸入名字、Email…等資料後按下Sign Up。會檢查Email的前綴是否有重複、密碼強度是否足夠、是否有空欄位，如果沒問題就可以創立成功。

#### 登入

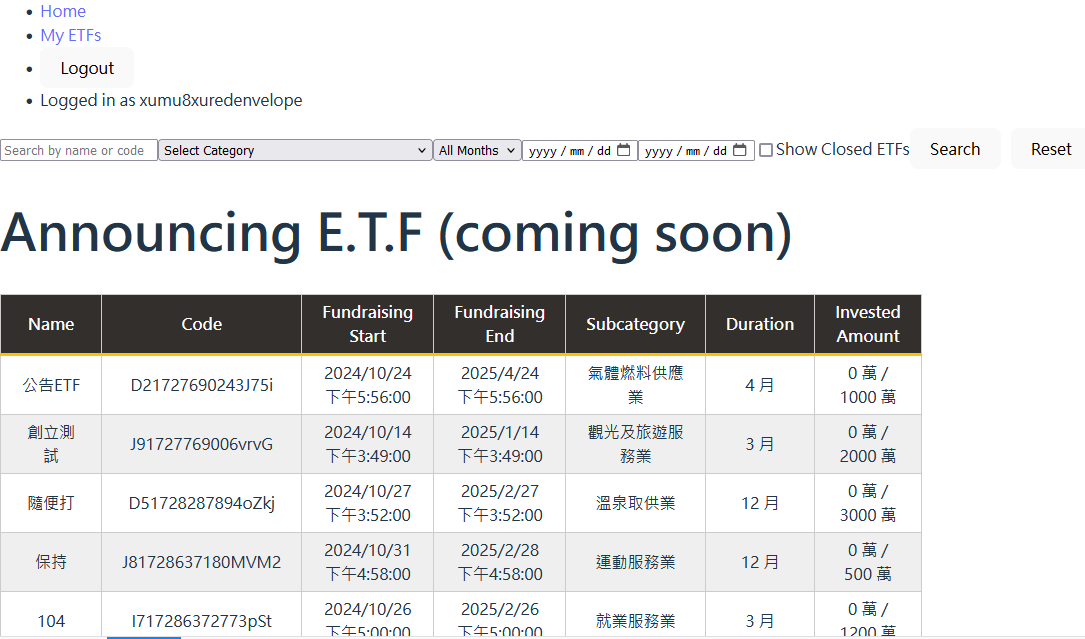
如果是首次登入，要先[創建帳號](#_創建帳號)，帳號創立完後會自動登入。從第二次起的登入則要進入登入頁面。



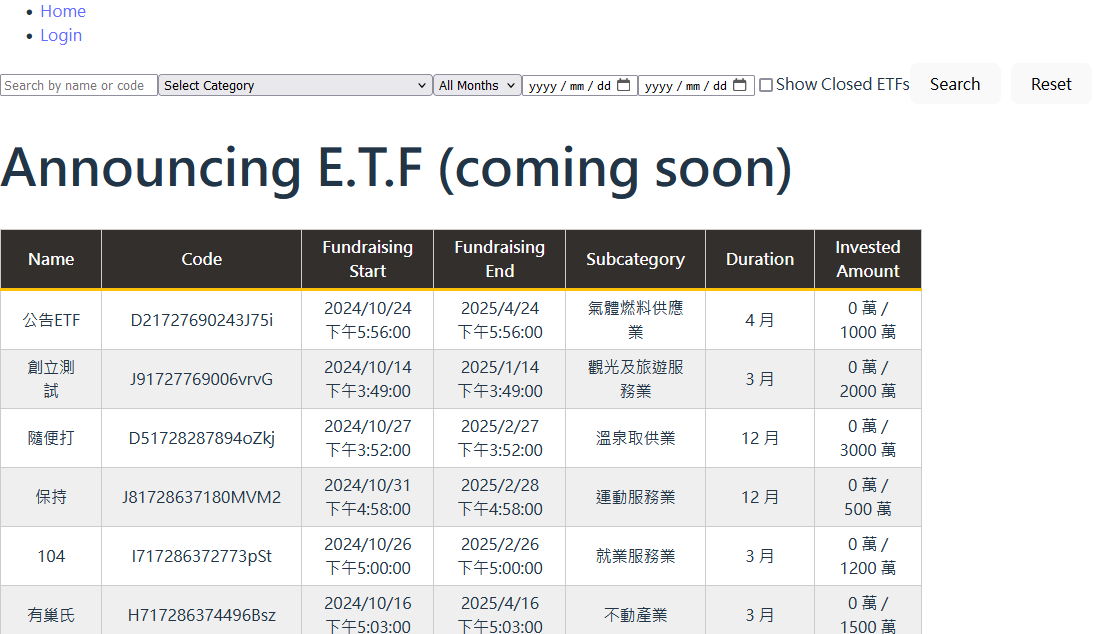
輸入Email和Password並按下Sign In，如無誤即可登入成功，並跳轉到首頁。

#### 登出

在登入狀態下，於任意頁面上方的導覽欄上可找到Logout按鈕（如下圖）。點下Logout按鈕即可登出，並跳轉到登入頁面。



### 檢視ETF列表（公告中、募資中）[未登入]





#### 頁面切換 Pagination

資料每10筆一頁，所以當資料超過10筆時，將有資料在後續頁面，此時在資料列下方會有Next Page按鈕，點擊可以進到下一頁（如上圖所示）。同理，當不在第一頁時，會有Previous Page按鈕，點擊可以返回上一頁（如下圖所示）。



#### Elasticsearch搜尋功能（[前端](vscode://file/C:/Users/user/Documents/yiyuan_exchange_platform_project/frontend/src/views/etfs/ETFSearch.jsx)、[後端](vscode://file/C:/Users/user/Documents/yiyuan_exchange_platform_project/backend/search/views.py)）

首頁上方的搜尋列包含了名稱＆單號搜尋、類別搜尋、ETF時長搜尋，以及時間區間搜尋。以上的所有搜尋都可以合併使用。此外，還可以選擇顯示募資額度已達成的ETF（預設隱藏，如募資500萬，達500萬或以上即達成募資額度）。

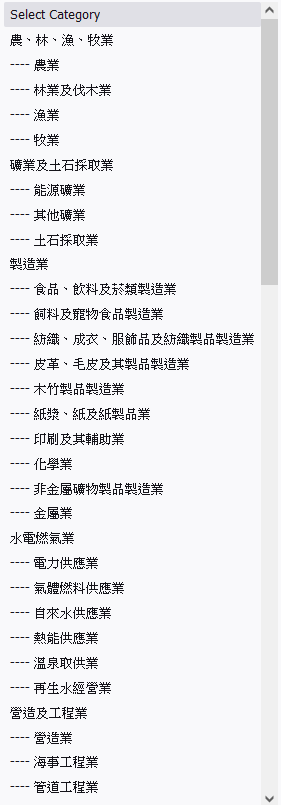
目前的搜尋功能或篩選功能只有首頁的ETF列表是用Elasticsearch實做，其他部份不確定有沒有這個需求，如果在userETF的頁面也要用Elasticsearch的話，需要改寫目前的程式碼。

##### 名稱&單號搜尋

ETF名稱為模糊搜尋，單號則為精確搜尋（連同大小寫也要正確），兩者只要有一者符合即可。

##### 類別搜尋

類別搜尋的類別資料，來源為統計資訊網的行業統計分類。共有12個主類別，78個次類別（如下圖）。次類別前綴有4個「-」，點選主類別可以看到該類別下的所有次類別資料，例如點選「礦業及土石採取業」，則能源礦業、其他礦業、土石採取業這三類的資料都會呈現。



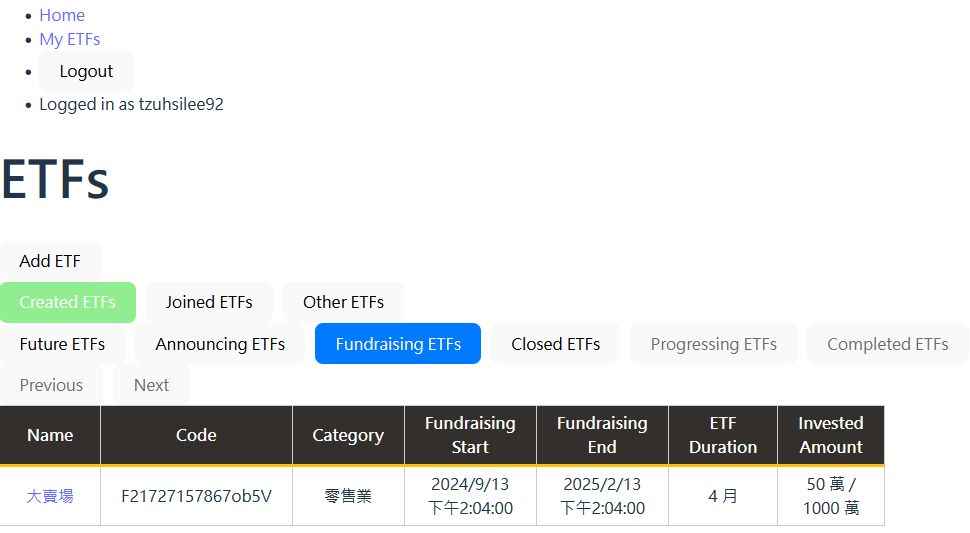
##### 時長搜尋

時長為精確搜尋，範圍為3~36個月，使用下拉式選單的形式。

##### 時間區段搜尋

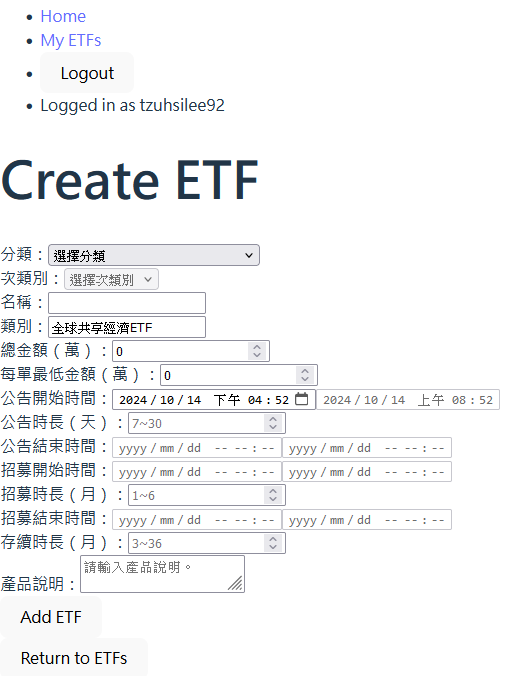
時間區段為精確搜尋，需填入兩個時間方可搜尋，且第一個時間需要比第二個時間早。只要時間區段內有涵蓋對應的起始時間或結束時間即成立。

### 管理ETF（創建、加入、其他…等等）[登入]



ETFs下的第一排按鈕Add ETF可以創建ETF，再之下的兩排是篩選條件，方便前端呈現資料。上排根據加入狀況區分，分為創建（Created）、加入（Joined）、其他（Other）；下排則是根據時間狀態區分，分為未來式（Future）、公告中（Announcing）、募資中（Fundraising）、進行中（Progressing）、已關閉（Closed）。這邊需注意的是，並非所有情境都存在，實際可能出現的狀態小於15個（=3\*5）。以下分別就這些部份進行說明。

#### 創建ETF



上圖為創建ETF時可以設定的選項，與[etfs/models.py](vscode://file/C:/Users/user/Documents/yiyuan_exchange_platform_project/backend/etfs/models.py)內的ETF model對應，詳細的變數名稱與資料型別可以在那邊找到。

這邊的分類為前文提到的主類別，來源一樣是統計資訊網的行業統計分類。這邊已經整理成JSON檔，選了分類後可以選擇下屬的分類別。此處與下方的類別「全球共享經濟ETF」有關，公益ETF、全球共享經濟ETF…等ETF屬於非商品類的ETF，具體有哪些非商品類的ETF及所屬的階層架構要再詢問詳細狀況。這邊保留的「類別」只是暫時留存，提醒有這個問題還沒處理；最後應刪除「類別」，並將非商品類別整理成與商品類別相同的格式後一併呈現。

總金額的限制，目前因為還沒有寫權限相關的系統，所以只有在[etfs/serializers.py](vscode://file/C:/Users/user/Documents/yiyuan_exchange_platform_project/backend/etfs/serializers.py)寫下限100（萬），之後把企業的部份補完後應根據個人或企業設定上下限。每單金額的限制目前也是下限2（萬），可再根據實際需求修改。

為了避免ETF處在奇怪的狀態（好比公告結束了卻還沒開始招募），公告時間與招募時間在設定上是連續的，亦即公告結束時間 = 招募開始時間。所以，只要給定公告開始時間和公告時長，就會得到招募開始時間；這邊再加上招募時長，就會得到招募結束時間。在目前的版本中local time和UTC time是並陳的，前者為local time，後者為UTC time，可再視情況取用需要的時間即可。

接下來要設定的是存續時長。這個是ETF時長，也就是錢會存在ETF池子裡面多久。最後可選的設定是產品說明，是向招募群眾說明這個ETF的內容，非必填。

目前版本中，如果時間輸入0的話，由於它是即時呈現修改，所以會出現Datetime Error（因為不存在0年、0月、或0日）。可以考慮使用Air Datepicker之類的方式改寫，或是在邏輯內增加判斷式處理0的Error。

下面分別根據各個加入狀況來說明其與ETF時間狀態互動的交互。

#### 加入狀況

##### 創建（Created）

顧名思義，這邊的ETF都是自己創建的。

ETF.objects.filter(creator=user)

需注意，自己創建的ETF，自己是不能加入的。

##### 加入（Joined）

同樣名稱解釋了一切。這邊是別人創建的ETF，而且自己有加入的。

ETF.objects.filter(useretf\_\_user=user)

透過UserETF，找出所有該用戶加入的ETF）。

##### 其他（Other）

既不是自己創建，也沒有加入的ETF都歸屬於此類。

ETF.objects.exclude(creator=user).exclude(useretf\_\_user=user)

由於是排除法挑剩下的，所以與前兩項的聯集就包含了所有ETF。

#### 時間狀態

##### 未來式（Future）

還沒有開始公告的ETF屬於此類。

etfs.filter(announcing\_start\_date\_\_gt=current\_time)

##### 公告中（Announcing）

目前正在公告中的ETF。

etfs.filter(

                announcing\_start\_date\_\_lte=current\_time,

                announcing\_end\_date\_\_gte=current\_time

            )

##### 募資中（Fundraising）

目前正在募資中的ETF。

etfs.filter(

                fundraising\_start\_date\_\_lte=current\_time,

                fundraising\_end\_date\_\_gte=current\_time

            )

##### 已關閉（Closed）

已經不再募資的ETF。

etfs.filter(fundraising\_end\_date\_\_lt=current\_time)

##### 進行中（Progressing）

目前正在進行中的ETF。

useretf.filter(leave\_date\_\_gte=now)

需注意的是，在現行版本中，只有加入的ETF有進行中的時間狀態，取代原本放在募資中的位置（進行中的ETF仍然可能有募資中狀態，只是不放在狀態列，而是在下表資料列中增加一欄Raising Fund）。這邊是預設使用者更在乎查看自己目前進行中的ETF，只有考慮是否要對已經加入的ETF增加資金投入時，才會參考是否仍在募資中。

##### 已結束（Completed）

與前者相同，同樣是只有進行中的ETF才有的狀態，標示目前使用者的該ETF是否已經結束，是相對於進行中的狀態。

useretf = useretf.filter(leave\_date\_\_lt=now)

如果募資時長超過ETF存續時間，好比募資時長為12個月，且ETF存續時間為3個月，則有可能在結束之後再次投入同一個ETF，進而重新變成進行中狀態。

需注意這邊和已關閉是不同的。對個人來說，關閉募資後，自己的ETF仍然可能在進行中，直到ETF時長結束或離開才是已結束狀態；對ETF來說，則是直到所有該ETF的用戶都結束才會變成已結束的狀態。