

AI 編程新時代：

Vibe Coding 零成本開發專屬App

▶ AI 給出的開發 App 所需的完整產品需求文檔 (PRD)。



3C 科技

在上一篇 (《大公報》6月20日3C科技版)，筆者结合自身3個月開發6個App的實戰經驗，帶大家整體了解了Vibe Coding (自然語言編程) 的概念和它給我們生活帶來的意義。筆者認為我們已經處於「人人都可編程」的時代。Vibe Coding是一種讓人欣喜的新體驗，筆者把它比喻為AI時代的「黏土」，任何人都可以用它把自己腦海裏的想法變成具體可見的軟件。為了讓大家也能切實體會到這種「造物」的樂趣，讓大家可以開始通過Vibe Coding嘗試解決遇到的困難，今天，筆者將通過一個實際案例帶大家一起從零到一用完全零成本的方式Vibe Coding一個App。

姚剛

需求準備

在正式開始Vibe Coding之前，我們需要提前做好一些準備工作，這樣有利於我們能夠在更短的時間內成功開發出真正可用的App產品。

第一步，確定需求

作為案例，今天，我們就用一個筆者切身的痛點作為這個App需要解決的問題。筆者生活中經常有一些需要周期性續費或繳費的事務，但往往因為沒有全部設置提醒而忘記繳費。「做一個可以定期循環提醒自已所有繳費事宜的iOS App」就是我們今天要完成的目標。

第二步，細化需求

在確定了我們要做的App的方向後，我們要用文字仔細描述並整理一下這個App應該有哪些具體的功能。下面就是筆者對今天要開發的這個循環提醒App的需求描述：我想做一個可以周期性提醒我進行不同繳費事項續費的iOS App。它可以輕鬆添加要提醒的付費事項；可以設置：事項的名稱、提醒周期、具體提醒時間、一些備註、重要度等。

第三步，先跟AI聊聊需求

在寫下自己整理的App功能需求後，筆者建議大家可以先找一個主流的AI (比如，ChatGPT 或者豆包等)，把你要做什麼樣的App告訴它，讓它幫你補充和完善一下需求，甚至可以讓它給出核心功能的參考代碼，以及一些用戶界面的設計要求。根據筆者經驗，這一步可以幫我們節省約一半的開發時間。

我們與AI對話，探討並完善這個App的需求後，AI會給出開發這個App所需的完整產品需求文檔 (PRD) 和一些可以用來給AI參考的核心代碼。AI甚至幫我們想好了這個App的名字——Renewly。



▶ 蘋果官方的iOS開發軟件「Xcode」。

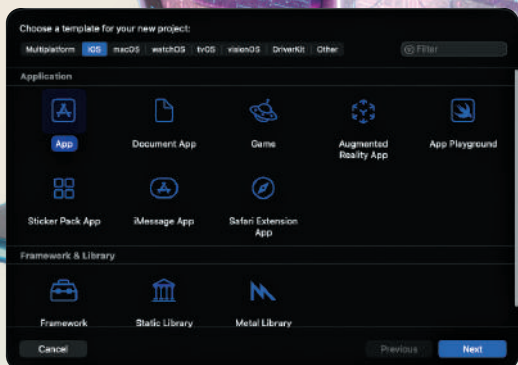
工具準備

接下來，在我們安裝和設置好所需的Vibe Coding工具後就可以正式開始App的開發環節了。今天，筆者帶大家嘗試開發的這個案例是一個iOS App。所以我們需要下載安裝兩個工具。一個是蘋果官方的iOS開發軟件「Xcode」；另一個，是筆者常用的由字節跳動推出的免費Vibe Coding工具Trae CN；建

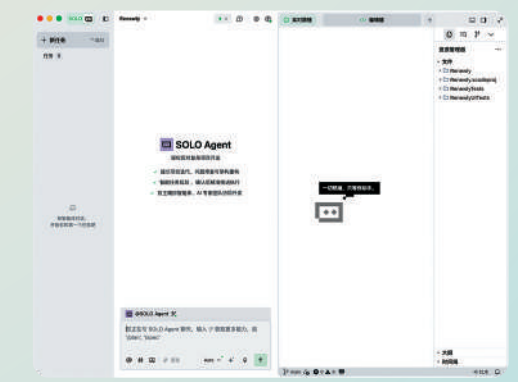


大公報AI製圖

▶ 開始Vibe Coding的第一步，選擇創建「iOS」(系統)下的「App」。



議安裝後打開solo模式，更適合新手。首先，我們需要理解一下這兩個工具的用途和關係。Xcode主要用來運行我們App內的代碼文件，而Trae則是實際撰寫這些代碼文件的工具。Trae撰寫所有的代碼文件會保存在Xcode創建的這個案例App的項目文件夾內。Trae負責寫代碼，寫完後，我們點擊Xcode裏的運行按鈕，就可以看到App是否能夠正常運行或出現了哪些bug (錯誤提示)。



▶ 第二步，打開Trae，點擊「打開文件夾」選中剛才Xcode創建的項目文件夾。

創建項目

開始Vibe Coding的第一步，是要創建這個App所屬的項目。因此，我們應該先打開Xcode，然後點擊「創建新項目」，選擇創建「iOS」(系統)下的「App」這個類型後，點擊「下一步」。

然後我們需要根據提示，填寫產品名稱 (建議使用純英文或數字)，其他選項可以保持默認，並繼續點擊「下一步」。

緊接着，Xcode會讓你選擇一個文件夾，用於存放這個項目的代碼文件。這個文件夾也是接下來我們使用Trae開始開發時需要使用的項目文件夾 (文件夾名稱建議使用英文，避免可能出現的報錯)。

第二步，打開Trae，點擊「打開文件夾」選中剛才Xcode創建的項目文件夾，我們就能在Trae看到剛才項目文件夾裏相同的基礎文件了。

以上兩步對於新手，可能會出現文件夾設置錯誤的問題，大家可以反覆嘗試，直至可以成功用Trae打開相同的項目文件夾。完成這一步，恭喜你，你已經完成了Vibe Coding的主要準備工作。下面，你就可以開始AI編程之旅了。

開始Vibe Coding

接下來，我們把最開始跟AI對話後，讓它幫我們撰寫的產品需求 (PRD) 內容和參考代碼複製黏貼到Trae的對話框裏，點擊發送，即可看到Trae開始根據你的需求制定開發計劃，並逐步開始撰寫代碼文件。在最右側的文件列表裏，你會看到一個個新的代碼文件的出現。這就是Vibe Coding的第一步，提出需求，讓AI開發對應的功能。

Trae類型的Vibe Coding工具在完成一些代碼的開發和改動後，會提示需要你進行文件的審查或確認。點擊確認，表示你認同這些代碼沒有問題。在實際操作中，如果你看不懂代碼也沒有關係，完全可以點擊接受。因為如果代碼有問題，在後續的App測試環節，會出現相應的報錯提示。屆時，我們再進行修復即可。

當Trae提示已經完成任務，代表它已經開發完成了你剛才發給它的那些需求。在向它提出新的需求前，筆者建議大家先進行一下模擬器測試，確認一下當前由Trae開發的功能有哪些問題。

我們需要回到Xcode，從頂部中間的「模擬器」下拉菜單中選擇一個常見的iPhone機型作為模擬器即可。然後，點擊左側欄上方的「運行」按鈕，即可看到代碼運行後有哪些錯誤。如果沒有嚴重錯誤，我們將可以順利看到Xcode打開了一個iPhone模擬器頁面。不過，首次運行時，往往會出現大量的報錯，這也是正常的。

運行後如果在Xcode裏左側展示出很多錯誤信息 (bug)，我們要做的是，先選中表示嚴重錯誤的紅色圖標的錯誤內容，右鍵複製這些報錯信息，並黏貼到Trae的對話框裏，告訴Trae：Xcode運行後有報錯如下，請修復。Vibe Coding的重要一環就是把運行後的



▶ 筆者與Trae經過5輪bug的修復，在模擬器上成功運行「循環提醒App」。

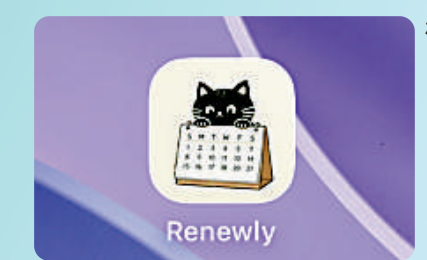
錯誤信息發給Vibe Coding工具，讓它逐步修復這些錯誤。在修復了大多數嚴重錯誤後，我們再次點擊Xcode上的運行按鈕。當我們看到「Build Success」的提示時，恭喜你，你將第一次通過模擬器看到你自已Vibe Coding的App的樣子。

我們本次作為案例的這個「循環提醒App」項目，筆者與Trae僅經過了5輪bug的修復，就在模擬器上成功運行了這個App。

在此基礎之上，如果你發現還需要增加一些功能或修改一些功能，我們可以繼續把你的想法通過自然語言的方式發送給Trae，然後繼續測試它開發出來的成果，如果有新的bug，就可以複製或截圖發送給Trae，讓它修復。以上就是真正核心的Vibe Coding過程。

應用圖標

為了讓這個App更加完整，你可以給這款App做一個應用圖標。方法也很簡單，我們可以找一個能生成圖片的AI工具，描述一下你需要生成一個什麼樣的圖標，並要求其尺寸是1024 x 1024像素。然後把這張圖片發送給Trae，告訴Trae，你希望將這張圖片用作這個App的應用圖標，請完成Xcode裏的相關配置。Trae就會根據蘋果官方的規範幫你完成配置。此時，再點擊運行按鈕，你將看到手機界面上出現了你的App圖標。



▶ Trae根據蘋果官方規範配置App圖標。

測試

開發一個可用的App，仔細測試每一個功能是不可忽視的重要環節。測試可以分為模擬器測試和真機測試兩種。筆者建議大家可以先在電腦上的模擬器中把所有功能都測試一遍，我們也可以給Trae下達一個指令：「請根據本App的功能生成關於這個App的完整測試用例文檔」，按照Trae提供的測試文檔，系統有序地進行測試，可以更全面地避免難以發現的bug。測試並確保沒有嚴重問題後，我們可以把自己的iPhone連接到電腦，此時，在Xcode上，我們剛才選擇模擬器機型的下拉菜單中，可以看到你的iPhone。選中你的iPhone，點擊「運行」按鈕，即可把你開發的這個App安裝到自己的手機上。這樣就可以開始「真機測試」了。

如果你只是打造一個自己用的App，那麼到這一步，你通過Vibe Coding開發的App作品就完成了。

寫在最後

今天介紹的步驟只是帶大家體驗通過Vibe Coding開發一個可以運行在自己手機上的iOS App的過程。如果大家有意將自己的App上架到蘋果App Store，後續還將面臨多個複雜的環節，如註冊蘋果開發者賬號、撰寫App介紹、製作App截圖、準備隱私條款、提交審核等。整套流程耗費的時間並不亞於前面Vibe Coding的過程。筆者將繼續跟大家分享這些複雜的上架流程。不過，如果你只是想體驗自己Vibe Coding開發一個App，按照筆者今天的分享，一步一步操作，一定可以實現你的這個目標。祝大家Vibe Coding愉快。