GitHub Pages 部署 Vue 專案詳解

本文件將詳細說明如何將 Vue 應用程式部署到 GitHub Pages 上。過程中每一步的操作都會做解釋，並提供背後的原因。  
部署 Vue 專案到 GitHub Pages 需要進行一些必要的配置，目的是讓專案能夠在 GitHub 上正確顯示，並避免一些常見的部署問題。  
以下是每個步驟的詳細解釋以及為何需要這麼做。

# 1. 建立 Vue 專案

首先，你需要創建一個 Vue 專案。使用 Vite（或 Vue CLI）來快速創建 Vue 項目，這樣能夠自動設置基本的專案結構和開發環境。  
  
為什麼這樣做：Vue 是一個前端框架，用來開發交互式網頁，而 Vite 是一個現代化的前端工具，專為快速開發和打包設計。這樣能讓你專注於編寫代碼，而不是手動設置開發環境。

# 2. 安裝 gh-pages 套件

gh-pages 是一個專門用來將靜態網站部署到 GitHub Pages 的工具。它可以幫助你自動將生成的靜態檔案推送到 GitHub Pages 的分支（通常是 gh-pages 分支）。  
  
為什麼這樣做：GitHub Pages 是 GitHub 提供的免費網站託管服務，並且只允許靜態網站，因此需要將靜態文件推送到特定的分支（gh-pages）。gh-pages 套件自動化了這個過程，讓部署變得簡單。

# 3. 設定 vite.config.js 中的 base 屬性

當你將網站部署到 GitHub Pages 上時，網址的結構會有所不同。如果你在 GitHub 上創建的是名為 my-vue-project 的倉庫，部署後網站的 URL 路徑會變成 https://yourusername.github.io/my-vue-project/。這樣，網站中的靜態資源（如 CSS 和 JS 文件）需要根據這個新的路徑來加載。因此，我們需要將 vite.config.js 中的 base 設定為 /my-vue-project/，以確保生成的靜態文件正確引用到 GitHub Pages 上的資源。  
  
為什麼這樣做：GitHub Pages 會將你的網站托管在一個子路徑下（例如 /my-vue-project/），如果不設置這個 base，Vite 生成的路徑會錯誤，導致靜態資源無法正確加載。

# 4. 推送專案到 GitHub

在將專案部署到 GitHub Pages 之前，你需要確保專案已經上傳到 GitHub 並與遠端倉庫相關聯。這是因為 gh-pages 套件會將生成的靜態檔案推送到 GitHub 上的特定分支。  
  
為什麼這樣做：GitHub Pages 是由 GitHub 托管的，它依賴於你推送的代碼來顯示你的網站。必須將專案推送到 GitHub，並確保設置好遠端倉庫，這樣才能正常使用 gh-pages 推送靜態資源。

# 5. 執行 npm run build 生成靜態文件

使用 npm run build 命令來生成生產環境的靜態文件。這會將你的 Vue 專案轉換成瀏覽器可以直接加載的 HTML、CSS 和 JavaScript 文件。  
  
為什麼這樣做：Vue 是一個動態的前端框架，必須將其編譯成靜態文件，這些文件才能部署到 GitHub Pages 上。npm run build 就是將 Vue 應用程序打包成靜態文件的步驟。

# 6. 執行 npm run deploy 部署到 GitHub Pages

當你執行 npm run deploy 時，gh-pages 套件會將 dist/ 資料夾中的靜態文件推送到 GitHub 上的 gh-pages 分支（如果還不存在會自動創建）。這樣，GitHub 就會將這些靜態文件顯示為網頁。  
  
為什麼這樣做：GitHub Pages 會從 gh-pages 分支加載靜態文件，因此，必須將打包後的靜態文件推送到這個分支，這樣 GitHub 才能夠提供你的網頁。

# 7. 檢查 GitHub Pages 網站

當完成部署後，你可以在 https://yourusername.github.io/my-vue-project/ 檢查你的網站。這是 GitHub Pages 預設的網址格式，專案會被托管在這個位置。  
  
為什麼這樣做：部署完成後，這是你可以在瀏覽器中查看你專案是否正確顯示的方式。

# 8. 設定 Vue Router 為 Hash 模式（可選）

如果你的 Vue 應用使用了 Vue Router，並且你在 GitHub Pages 上遇到 404 錯誤，則需要將 Vue Router 的路由模式設為 hash 模式。這樣，路由會使用 URL 中的 # 符號來指示不同的頁面，避免了 GitHub Pages 無法處理路由的問題。  
  
為什麼這樣做：GitHub Pages 是一個靜態網站託管服務，當你使用 Vue Router 的 history 模式時，可能會遇到 GitHub Pages 不能正確處理路由的問題，這時使用 hash 模式能夠解決這個問題。

# 總結

這些步驟的目的是為了讓你的 Vue 應用能夠在 GitHub Pages 上正確運行並展示。從配置 base 路徑到使用 gh-pages 部署靜態文件，這些操作確保了你的專案能夠順利上線並正常運行。每一步都是為了讓應用程序在 GitHub Pages 上被正確加載，並解決了靜態網站部署過程中的常見問題。