

作業說明

github page: <https://github.com/eethan1/myWebSite>

這個作業我實作了 1.限制 一段時間內的請求次數 和 5.photouploader 後端

整個網頁前端無框架，沿用 個人網站和handlebars的樣式

後端採用 nodejs 搭配 express 框架

資料庫採用 MongoDB 搭配 Redis

我將整個和這次 Dcard 相關的東西打包成 dcarso.js 和 dcarsodb.js，方便在 index.js 中快速啟用

後端頁面架在個人網頁的基礎上，可以直接在 [yggd2mir](#) 查看實際狀況。

在實際跑專案時，需要將 *-default.js 的 config 檔案設定後改為 *.js

hw1 :

我在第一次很快的使用 mongoose 建立初版本的中間件後，發現同一頁面多個請求會造成資料庫 race condition，我使用 async.queue 避免 race condition 和 redis 快取避免請求時間太長。相關的程式碼可以在 dcarso.js 和 dcarsodb.js 中找到， config 檔案在 ARG-default.js 中。

hw5:

我使用 Sharp 作為圖形處理庫，並用 multer 作為上傳的中間件。

這題花較多心力在處理安全方面，我設定了檔案上傳大小、檢查上傳副檔名；使用 buffer 到記憶體後用 file-type 確認檔案實際類型；回傳縮圖時過濾 . 和 / 等等。

相關程式碼在 dcarso.js 中， config 檔案同樣在 ARG-default.js 中

/photoUploader

簡單的上傳頁面

GET /api/img/:imgid?w={width}&h={height}

指定長和寬還有圖片名稱後回傳圖片

POST /api/img

上傳圖片後回傳

{

status: Number http status code,

filepath: String 檔案路徑,

type: String mimetype

}

透過 filepath access 上傳後的檔案