



nhentai

@nhentaiOfficial

nhentais goal is to organize the worlds hentai and make it universally accessible and fappable

[翻譯自我介紹](#)

📍 The 9th Circle of Hell [🔗 nhentai.net](#) 📅 已加入 2014年6月

37 個跟隨中 3.5萬 位跟隨者

未被你跟隨的任何人跟隨

期中專題

視窗元件應用練習

題目：

NHENTAI DOUJINSHI PREVIEW AND DOWNLOADER

姓名 余明霖| 學號 108021063| 課程名稱 視窗程式設計| 日期 2020/11/09

動機與目的

原本是想用一樣的手法對另一個網站 [Pixiv.net](https://www.pixiv.net/) 做類似的工具的。但經過我仔細的深入研究發現製作時程可能趕不及期中報告交件的時間，於是臨時更換了目標網站改做 nhentai.net 這個成人網站的工具。對於這類軟體的開發動機可以用一句話來概括，也就是「科技始終來自於人性」，對於人們而言，娛樂的需求是不可視而不見的。也因此我想透過這類創意的簡單實作，試著激發人們對於提供創新娛樂方法的想像力。

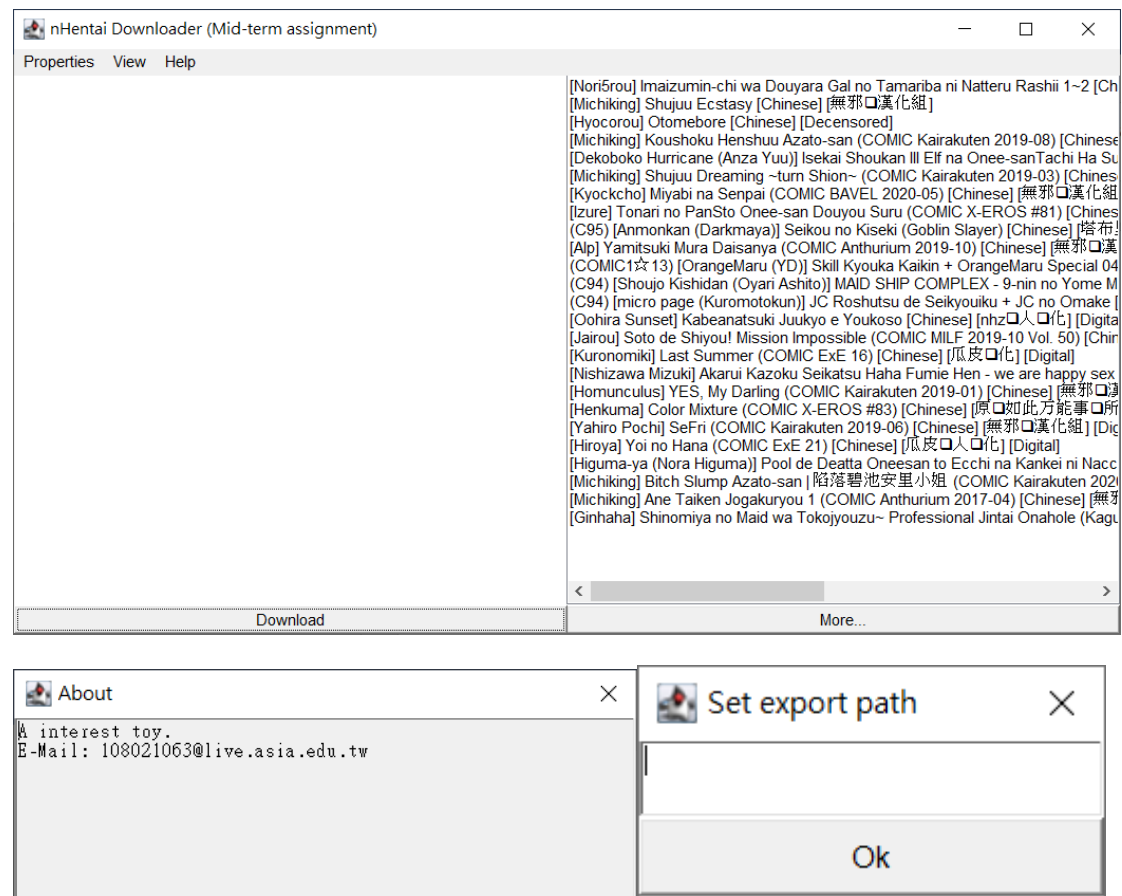


這個軟體發想於美國選戰期間，各大社群網站「烽火連天」。就只剩下少數網站上還可以維持寧靜友善，在逃離肅殺的社群網站論戰時，突然想到的好點子。

關於這個軟體的實作，大略先以實現已下功能為重點。

- 預覽同人本
- 下載同人本
- 以 ID 下載同人本

專題使用視覺元件與事件



- Frame、Dialog 元件 用於開啟視窗和各種小視窗
- onCloseing 事件 點右上角關閉可以正常關閉視窗
- Window 用於顯示圖片預覽
- List 顯示作品列表
- ActionListener 左鍵雙擊選項或者觸發 Enter 按鍵時顯示圖片預覽於左側
- Button 顯示按鈕
- ActionListener 各種按鈕功能

參考資料、使用的 Lib 與我自己創新的設計

參考資料和使用的 LIB

A Visual Guide to Layout Managers

<https://docs.oracle.com/javase/tutorial/uiswing/layout/visual.html>

Lesson: Concurrency

<https://docs.oracle.com/javase/tutorial/essential/concurrency/threads.html>

Learn Kotlin

<https://kotlinlang.org/docs/reference/>

OkHttp Recipes

<https://square.github.io/okhttp/recipes/>

Gson

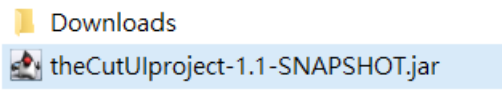
<https://github.com/google/gson>

創新設計

沒有什麼特別的創新設計，就是很正常的完成一支程式。

像是在遇到 I/O Blocking 的時候開 Thread、等 Http 超級慢的回應的時候開 Thread，利用 Kotlin 節省開發壓力等等...

執行過程與結果



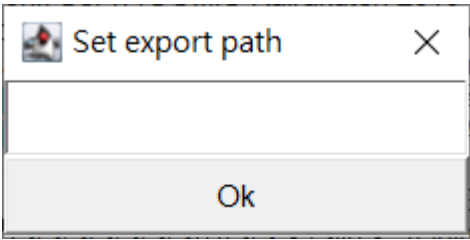
自動建立存檔資料夾



簡單同人誌預覽



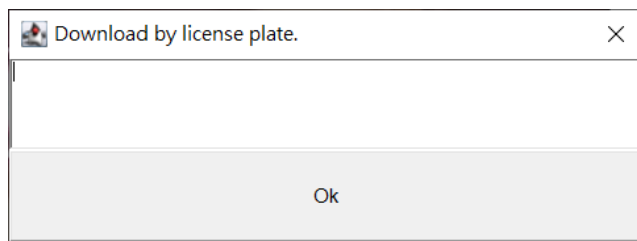
顯示更多同人誌



更改下載位置



下載同人誌



用 ID 下載同人誌



關於視窗

程式碼解說、討論與問題

程式碼摘要

```
fun search(keyword: String, page: Int, sortedPopular: Boolean = true): JsonObject {
    val searchUrl: String = if (sortedPopular) {
        "$searchPrefix$keyword&page=$page&sort=popular"
    } else {
        "$searchPrefix$keyword&page=$page"
    }

    val request: Request = Request.Builder()
        .url(searchUrl)
        .addHeader(headerName, headerValue)
        .build()

    client.newCall(request).execute().use { response ->
        return JsonParser.parseString(response.body!!.string()).asJsonObject
    }
}
```

Query 搜尋結果 String 轉 Json 然後 return

```
fun getInfo(ID: String): JsonObject {
    val request: Request = Request.Builder()
        .url(url: "$infoPrefix$ID")
        .addHeader(headerName, headerValue)
        .build()

    client.newCall(request).execute().use { response ->
        return JsonParser.parseString(response.body!!.string()).asJsonObject
    }
}
```

Query 單 ID 的結果轉 Json 然後 return

```

fun getThumb(media_id: String, type: String): InputStream {
    val request: Request
    if (type == "j") {
        request = Request.Builder()
            .url(url: "$thumbPrefix$media_id/thumb.jpg")
            .addHeader(headerName, headerValue)
            .build()
        return client.newCall(request).execute().body!!.byteStream()
    } else if (type == "p") {
        request = Request.Builder()
            .url(url: "$thumbPrefix$media_id/thumb.png")
            .addHeader(headerName, headerValue)
            .build()
        return client.newCall(request).execute().body!!.byteStream()
    }
    return InputStream.nullInputStream()
}

```

Thumbnail 轉成 InputStream 然後 return

```

fun getImage(media_id: String, img: Int, type: String): InputStream {
    if (type == "p") {
        val request: Request = Request.Builder()
            .url(url: "$imagePrefix$media_id/$img.png")
            .addHeader(headerName, headerValue)
            .build()
        return client.newCall(request).execute().body!!.byteStream()
    } else if (type == "j") {
        val request: Request = Request.Builder()
            .url(url: "$imagePrefix$media_id/$img.jpg")
            .addHeader(headerName, headerValue)
            .build()
        return client.newCall(request).execute().body!!.byteStream()
    }
    return InputStream.nullInputStream()
}

```

圖片轉 InputStream 然後 return


```
init {  
    createAndShowGUI()  
    refreshComicList()  
    checkDict(savePath)  
    this.isVisible = true  
}
```

視窗主要就是創建一個 GUI，接著更新列表，最後開始創建儲存下載資料的資料夾，最後顯示出視窗。

完成。

問題與討論

為了趕在期限前完成可能在錯誤處理、輸入檢查方面有潛在的漏洞，多執行緒的處理更是有些潦草以及蘊含許多思慮不周的部分。情況允許應該儘可能修復完善。

然後 Windows 該死萬年預設 Big5 過氣時代眼淚編碼到底什麼時候才要換！？

心得

花最多時間的事情在了解 nhentai.net 網站既公開但卻又沒有文件的 API 應該怎麼呼叫並且嘗試導入視窗中完成這次專題。另外，對於使用者介面的設計實在不是我心之所向...怎麼做怎麼醜讓我有些挫折以外，過程裡修修補補然後一遍又一遍整理整個視窗裡的東西然後重新檢查一次視窗、每個按鈕戳過確定正常之類的流程實在是煎熬。堅持只用 AWT 完成整個介面的部分更是對 Try and error 以及 Try and cry 有了更完整的體驗，看著程式還是一堆 Method 的時候一切都很好，但 Run 完發現和自己想像中的以及文件裡寫的東西有所落差。在崩潰的路上還是得想些辦法 Work round 交件的體驗實在奇妙。