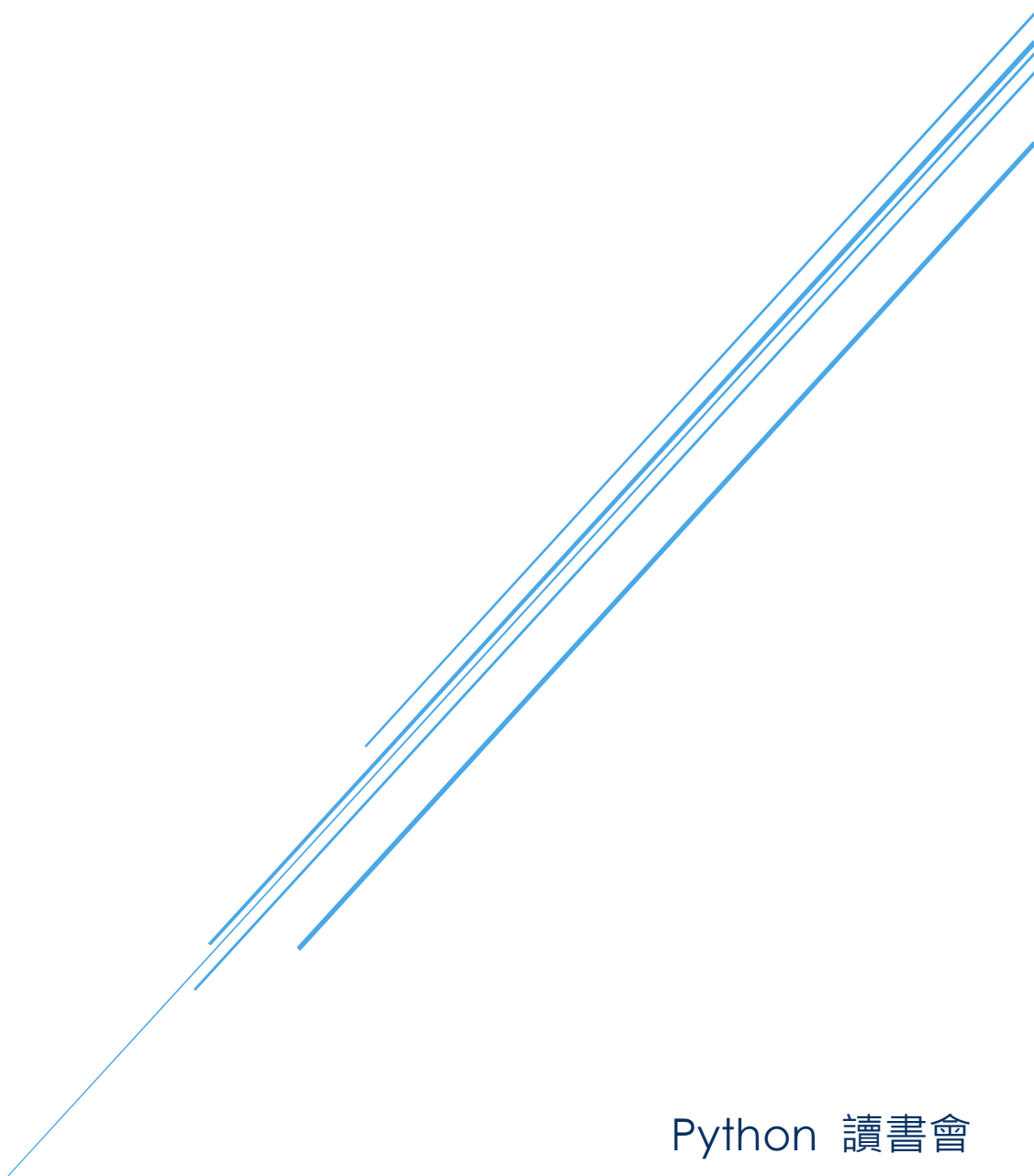


MANPOWER 計算系統

使用 JIRA Restful API 解決方案



Python 讀書會

Jerry & Van

版本紀錄

版本	內容	發佈日期	作者
v0.0.1	ManPower 軟體架構簡介	2024/09/17	Van
v0.0.2	Man Power tool 軟體需求	2024/10/08	Jerry

目錄

第 1 章 概論.....	4
第 1.1 節 Man Power 簡介.....	4
第 1.2 節 軟體開發環境	5
第 1.3 節 Http 和 Restful API	7
第 1.3.1 節 HTTP 協定.....	7
第 1.3.2 節 RESTful API	9
第 1.3.3 節 Man power tool 環境	12
第 2 章 Man power Tool 軟體需求	13
第 2.1 節 Man Power tool 介面	13
第 2.2 節 介面內容	14
第 2.2.1 節 介面 - 計算方式.....	14
第 2.2.1 節 介面 - 顯示資訊	15
第 2.2.1 節 介面 - 詳細資料選單	16
第 2.2.1 節 介面 - 檔案匯出	18
第 3 章 Man power Tool 軟體架構簡介	20
第 3.1 軟體設計架構	20
第 3.2 節 軟體套件簡介	22
第 3.2.1 節 Python PyQt5 套件	22
第 3.2.2 節 Python json 套件	22
第 3.2.3 節 python Requests 套件.....	22
第 3.2.4 節 python ini-parser 套件	22

第 3.2.5 節 python Jira 套件.....	23
第 3.2.6 節 python openpyxl 套件.....	23
第 3.2.7 節 Python matplotlib 套件.....	23
第 3.3 節 Man Power Tool 初始化執行流程.....	24

第 1 章 概論

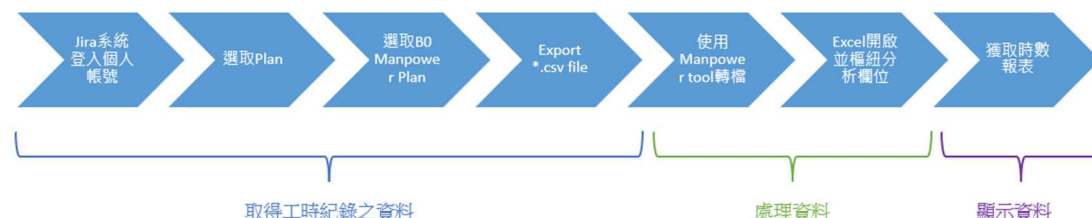
第 1.1 節 MAN POWER 簡介

Man Power 是一套 A 公司 用來計算人力工時的簡易程式；主要方便觀察同仁每周工時，是否有**超過或不足**之情況，以便供同仁自行調整。該程式的運作流程如下：



從上面的流程架構圖來看，使用者必須**手動**進入 AUO 的 JIRA 系統，選擇自己的計畫(Plan)並且從 JIRA 中下載其 CSV 檔案(裡面有 BU 上所有成員在不同專案花費的時間)。下載完 CSV 檔案後，同仁便可以利用 Man Power Tool，將下載的 CSV 檔案轉換成 Excel 文件。最後，同仁便可以**手動**改成較容易閱讀的形式(如**樞紐分析**)，並且得知自己與別人每周的工時。

因此，不難發現這一連串的動作中，很多動作仍然需要使用者使用「**手動**」的方式，個別完成。從這點出發，我們可以重新分析一下整個流程，如下圖：



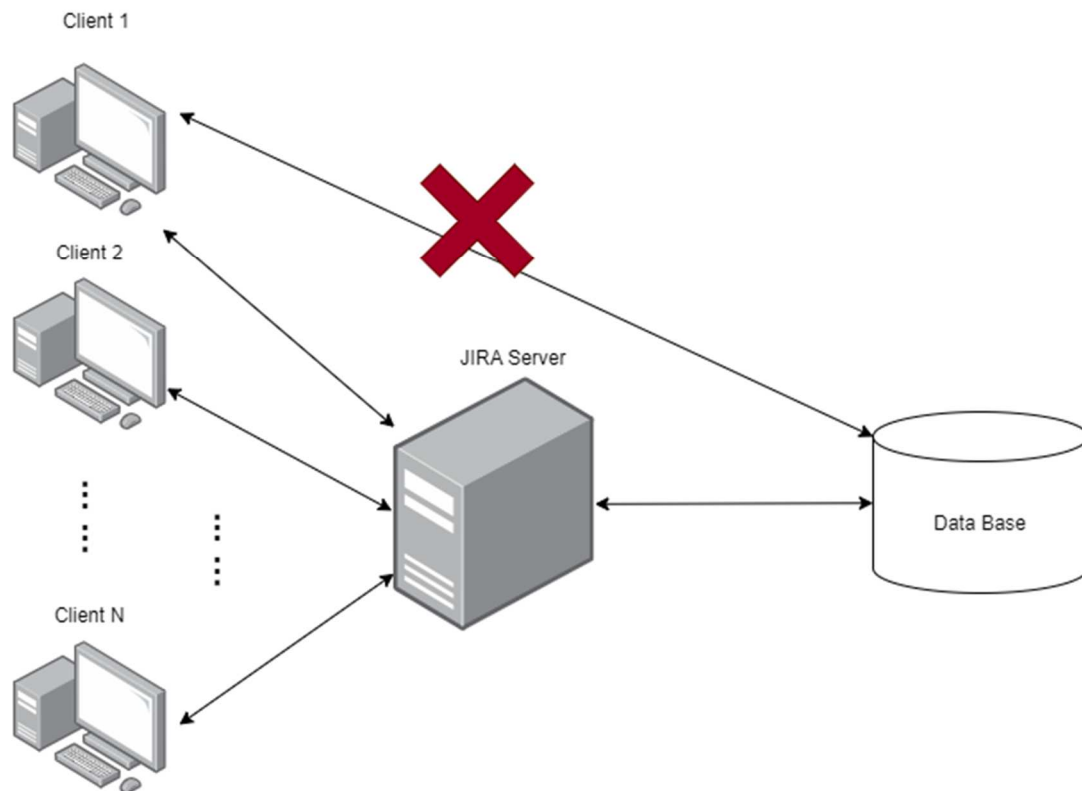
我們可以把整個流程分成三大部分，

- (1) 取得工時紀錄之資料；
- (2) 處理資料；
- (3) 顯示資料；

而本次 Man Power Tool 工時統計計算程式的「**重點核心**」也在於「取得工時紀錄之資料」，我們會採用網際網路常用的技術「**HTTP**」協定與「**RESTful API**」。

第 1.2 節 軟體開發環境

要實現 Man Power Tool，我們首先必須要克服一個難題；那便是如何存取 AUO JIRA 伺服器，儲存在資料庫(Database)的資料呢？因為我們客戶端是無法直接取得 AUO JIRA 伺服器背後的資料庫伺服器，如下圖：



由於我們無法直接存取 AUO JIRA 伺服器的資料庫，因此只能採取「**間接**」存取的方式，進一步對 AUO JIRA 伺服器的資料庫進行更新以及讀取；也就是說，我們必須透過 AUO JIRA 伺服器，間接地更新以及取得資料庫的資料。

常見的方式有二，一是採用爬蟲的方式，透過剖析伺服器回應的網頁取得我們需要的數據，以及錄製網頁對於伺服器的請求，進而更新資料庫的數據；這個方法聽起來就非常的麻煩吧！另一個方法則是透過**伺服器提供**的 HTTP API 或者是 RESTful API 來間接存取 AUO JIRA 伺服器資料庫；而 HTTP API 和 RESTful API 的差異，在下一節會介紹。第二個方式，聽上去雖然簡單，但還是需要伺服器本身有提供，才能讓客戶端有辦法透過 API 來更新與取得資料庫的資料。

必須強調的是，不管是方法一還是二，兩者使用的技術都是基於 HTTP(HyperText Transfer Protocol)協定，或說兩種方式都是利用 HTTP 協定傳輸資料，只是採用的技巧有所不同。我們可以這樣理解，當伺服器本身沒有提供 HTTP API 時，我們便只能採用爬蟲的方式以及錄製網頁對伺服器發動 HTTP 請求去存取資料庫的資料。

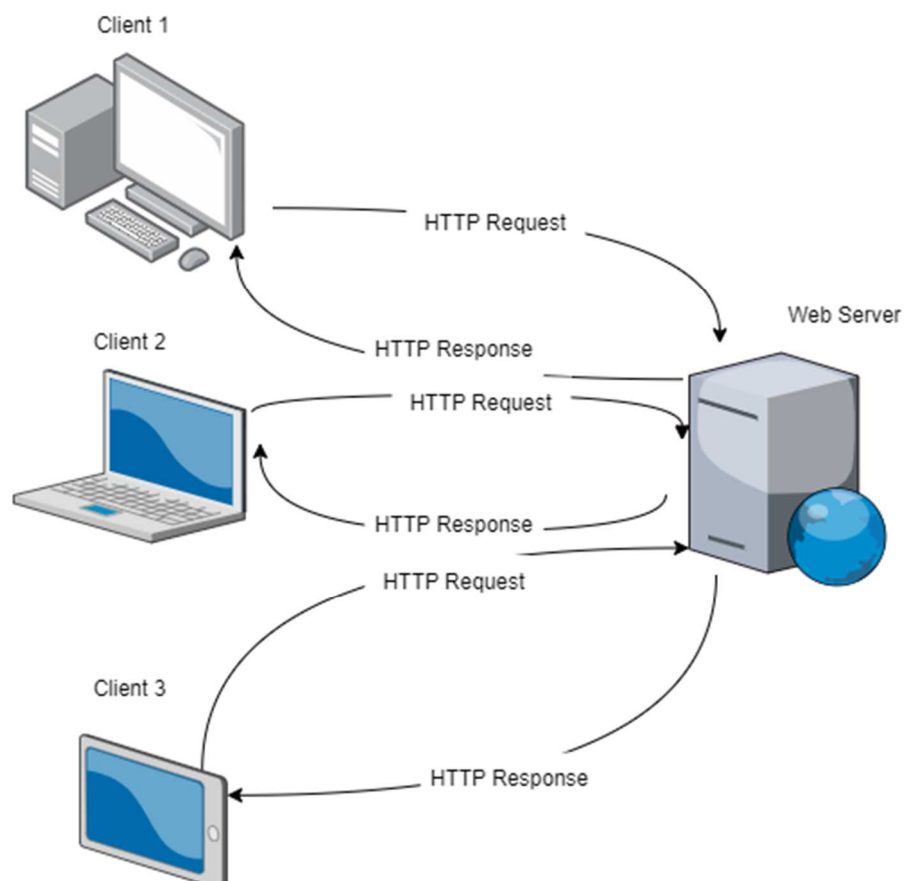
所幸，JIRA 伺服器有提供 HTTP RESTful API，我們可以利用 HTTP RESTful API 去存取 AUO JIRA 伺服器的資料庫。因此，在下一節，我們將簡介 HTTP 協定以及 RESTful API，然後進一步勾勒出我們軟體開發的環境。

第 1.3 節 HTTP 和 RESTFUL API

本節又分三小節，分別介紹 HTTP 協定、RESTful API 與 Man Power Tool 開發環境的介紹。

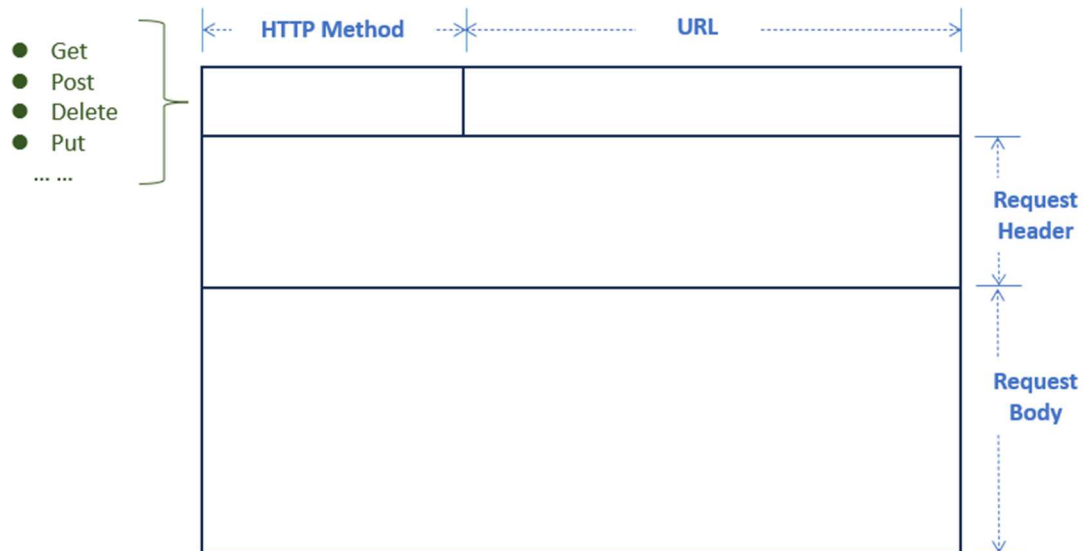
第 1.3.1 節 HTTP 協定

在網際網路的世界中，網站伺服器(Web Server)主要使用的通訊協定便是應用非常廣泛的 **HTTP(HyperText Transfer Protocol)**協定，**HTTP** 協定採用**請求(Request)/回應(Response)**機制；**請求**的意思便是對網站伺服器請求「**資源(Resource)**」，而網站伺服器接收到該請求以後，如果該請求伺服器允許的，便會給予對應的**回應**。如下圖：



HTTP 協定的誕生並不難以理解，畢竟網站伺服器需要與各式各樣不同的客戶端(Client)溝通，交換資源；為了避免溝通發生歧異，便需要統一的規範與通訊格式，HTTP 協定便定義了這一套客戶端與伺服器端之間溝通的方法。

既然 HTTP 協定如此重要，接下來，我們便需要簡單地了解一下 HTTP 請求與回應封包的格式；首先，我們先來認識一下 HTTP 請求封包，如下圖：



其中 HTTP Method 欄位常見的為 Get、Post、Delete 以及 Put，用來決定對網站伺服器進行的操作。URL 欄位則是填寫網站伺服器的位址。URL 的格式如下：

使用的協定://域名(or IP)[:port]/路徑(path)

例如：

<http://192.168.1.223:8080/test>

<http://localhost:8080/test>

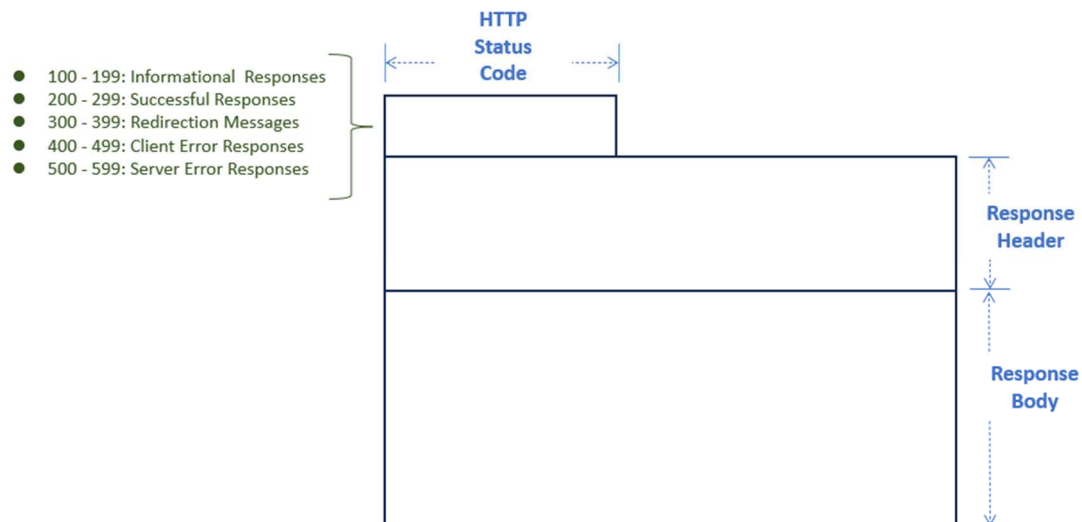
若沒有填寫埠號(Port)，預設為 80。Request Header 欄位可以讓使用者添加其額外的設定給予後端伺服器，如 Cookie。當 HTTP Method 為 Get 時，參數是放置在符號，"?"後面，例如：

<http://localhost:8080/test?id=124&name=Judy>

參數的格式為 Key=Value；多個參數時用"&"符號連接。若 HTTP Method 為 Put 或 Post 時，參數不會放置在 URL，而是放置在 Request Body 欄位之中，格式可以是 Key=Value 格式，

但若為 RESTful 風格，則為 XML 格式或是 JSON 格式。

HTTP Response 的封包格式則如下圖：



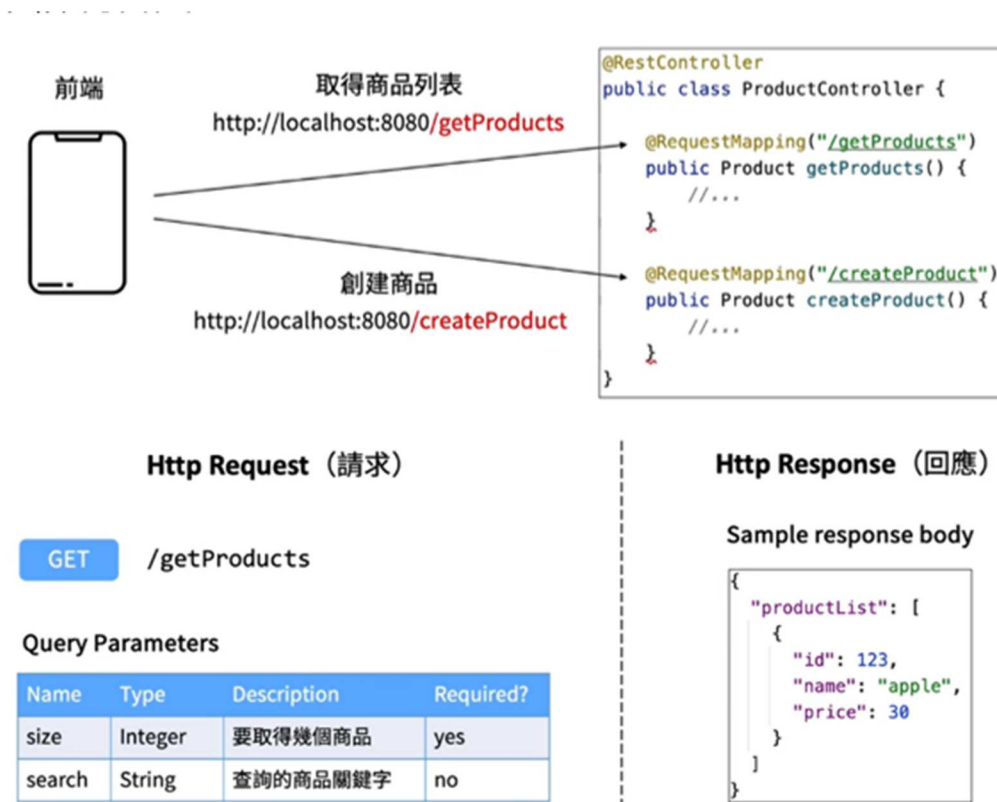
網頁伺服器收到請求後，會回應「請求的結果」，最直接明瞭的就是透過 HTTP Status Code 回應。我們不詳細介紹這個 HTTP Response，因為畢竟這些回應與實際的伺服器實作有很大的不同。更多有關 HTTP 的知識可以自行去網路搜索，我們就只將 HTTP 協定的基礎介紹到這裡。

第 1.3.2 節 RESTFUL API

在介紹 **RESTful API** 之前，我們先得解釋一下什麼是 **API**，所謂的 API 指的是：

1. API 是指用工程師的方式去說明某個功能的使用方法；
2. 目的：寫清楚這個功能要怎麼用；

而 **HTTP API**，自然就是要告訴開發前端或客戶端的工程師，要如何發送 HTTP 請求才可以得到或說達到伺服器提供預期的服務或是資源。如下頁的圖，



上圖提供 GET /getProducts，便代表前端或客戶端只需要發送 HTTP Method 為 GET，URL 路徑為 getProducts，且必須在"/"後面帶 size 參數的 HTTP 請求，便可以取得指定 size 的產品資訊。而所謂的 RESTful API 便是符合 REST 風格的 HTTP API。

何謂 **REST(Representational State Transfer)**？按照英文的翻譯，即**表現層狀態轉換**；這是甚麼意思呢？在網際網路的世界中，所有的一切統稱叫作「**資源**」(Resource)，而所謂的表現層狀態，指的便是「資源」的狀態。那麼如何表現出資源的狀態呢？一種直觀的想法便是利用 HTTP Method 去表達、表現現在想要如何處理「資源」。如下表：

HTTP Method	對應的資料庫操作	說明
POST	Create (新增)	新增一個資源
GET	Read (查詢)	取得一個資源
PUT	Update (修改)	更新一個已經存在的資源
DELETE	Delete (刪除)	刪除一個資源

如上表，我們便可以使用 HTTP Method 的 POST、GET、PUT 以及 DELETE 去**表現**請求資源的狀態，要新增還是要刪除等等之**轉換**。

但是還有個問題，我們要對什麼「資源」進行處理呢？為了讓 HTTP 請求可以**表現**出來這個狀態，**REST** 風格採用了 URL 路徑來描述資源之間的關係，如下表範例：

Http method + URL 路徑	說明
GET/users	取得所有 user
GET/users/123	取得 user id 為 123 的 user
GET/users/123/articles	取得 user id 為 123 的 user 所寫的所有文章
GET/users/123/articles/456	取得 user id 為 123 的 user 所寫的、article id 為 456 的文章
GET/users/123/videos	取得 user id 為 123 的 user 所錄的所有影片
GET/users/123/videos/789	取得 user id 為 123 的 user 所錄的、video id 為 789 的影片
GET/users/100	取得 user id 為 100 的 user

最後，則是傳輸資料之間的格式能夠統一，因此 REST 風格要求資料必須以 XML 格式或是 JSON 格式。

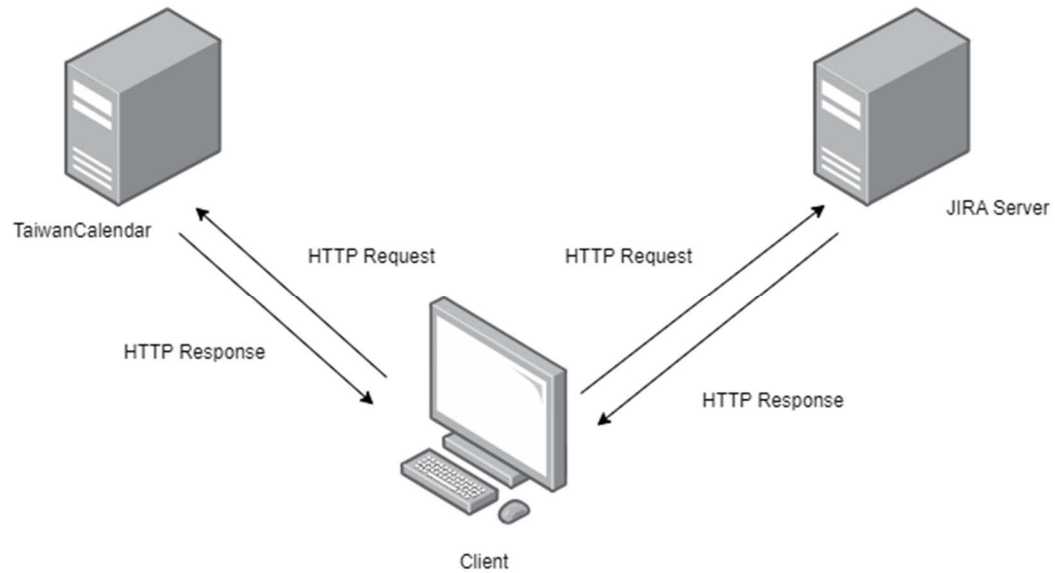
總結一下 RESTful API 的準則如下：

1. 使用 **HTTP** method 表示動作；
2. 使用 **URL** 路徑描述資源之間的階層關係；
3. response body 返回 json 或是 xml 格式；

RESTful API 沒有標準答案，永遠記住 RESTful API 實現的目的便是為了簡化團隊溝通的成本，我們本小節的介紹旨在助於我們能更好理解 JIRA API 設計，更多 RESTful API 的知識還是可以透過各式各樣的網路文件得到更進一步的認識。

第 1.3.3 節 MAN POWER TOOL 環境

Man Power Tool 的環境如下圖：



由於我們需要取得台灣的行事曆，所以 Man Power Tool 所在的客戶端，除了需要向 AUO JIRA 伺服器取得與更新資料外，也需要向 Taiwan Calendar 的伺服器取得目前年份的行事曆。

第 2 章 MAN POWER TOOL 軟體需求

第 2.1 節 MAN POWER TOOL 介面

依據需求，規劃樣式如下



1. 顯示登入狀態:

“已登入”：表示 Jira 已登入並連線成功(綠色)

“未登入”：表示 Jira 尚未登入或連線失敗(灰色)

2. 選擇計算方式:

原始計算：為保留與現行版本相同的計算方法

修正版：為扣除假期計算方法的版本

3. 顯示部門資料: 登入系統後，自動顯示該登入者所屬部門

4. **生成及更新資料:** 顯示部門時數資料於瀏覽修訂區域
5. **瀏覽修訂區域:** 瀏覽部門時數資料
6. **檔案匯出:** 匯出完整資訊檔案

第 2.2 節 介面內容

對於 Man Power Tool 介面做詳加介紹

第 2.2.1 節 介面 - 計算方式

因導入此軟體初期於推廣階段，故保留原本時數的計算方式與修正版計算方式，避免部門在溝通之間有所落差

原始計算: 依照填寫的時數及天數計算，只會計算工作天及假日去除，工作日遇上國定假日則無法

原始計算: 依照填寫的時數及天數計算，只會計算工作天及假日去除，工作日遇上國定假日則無法去除

Ex:

Project A
Date: 7/18 ~ 7/24 (假設7/19國定假日)
時數: 20 hrs

扣除假日均分時數至工作周

428: 08 hrs

429: 12 hrs

(若工作日遇上國定假日，仍視為工作日計算)

2024年		<	7月	>	放假安排		返回今天	
日	一	二	三	四	五	六		
30 廿五	1 廿六	2 廿七	3 廿八	4 廿九	5 三十	6 小暑		
7 初二	8 初三	9 初四	10 初五	11 初六	12 初七	13 初八		
428 14 初九	15 初十	16 十一	17 十二	18 十三	19 十四	20 十五		
429 21 十六	22 大暑	23 十八	24 十九	25 二十	26 廿一	27 廿二		
28 廿三	29 廿四	30 廿五	31 廿六	1 廿七	2 廿八	3 廿九		

修正版: 依照填寫的時數及天數計算，計算工作天及去除假日與工作日遇上的國定假日

Ex:

Project A
Date: 7/18 ~ 7/24 (假設7/19國定假日)
時數: 20 hrs

扣除假日均分時數至工作周，且自動扣除國定假日

428: 05 hrs

429: 15 hrs

2024年

<

7月

>

放假安排

返回今天

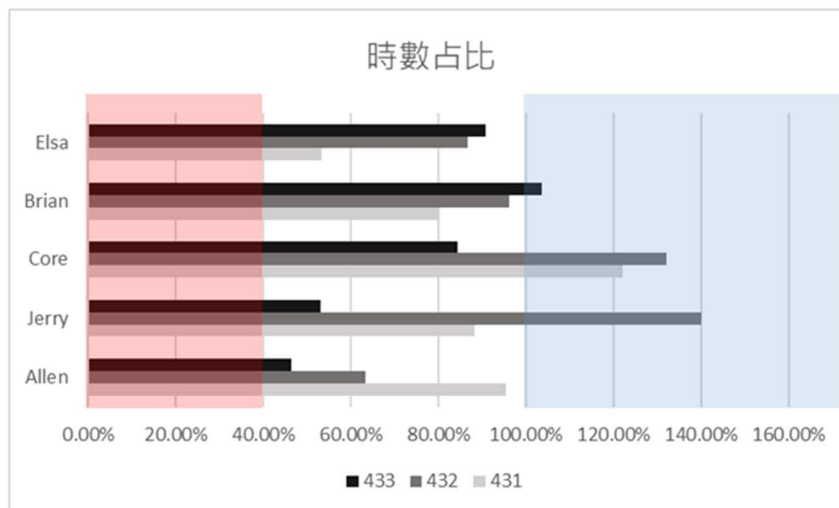
日	一	二	三	四	五	六
30 廿五	1 廿六	2 廿七	3 廿八	4 廿九	5 三十	6 小暑
7 初二	8 初三	9 初四	10 初五	11 初六	12 初七	13 初八
428 14 初九	15 初十	16 十一	17 十二	18 十三	19 十四	20 十五
429 21 十六	22 大暑	23 十八	24 十九	25 二十	26 廿一	27 廿二
28 廿三	29 廿四	30 廿五	31 廿六	1 廿七	2 廿八	3 廿九

第 2.2.1 節 介面 – 顯示資訊

- 依部門人員羅列出 “當週+後兩週” ，合計共三週的資訊
- 將分成三種顏色區分時數狀態，據當周工時所需的 60%及 100%做區分
 - 黑色: 符合當周所需時數($60\% \leq \text{工時} \leq 100\%$)
 - 藍色: 超過所需時數($\text{工時} > 100\%$)
 - 紅色: 少於所需時數($\text{工時} < 60\%$)
- 於部門時數資料區點選個人資料，可進入詳細選單內修改調整時數
- 時數占比以圖示顯示週工作時數的百分比 ((計畫工作時數/當週工作時數)*%)

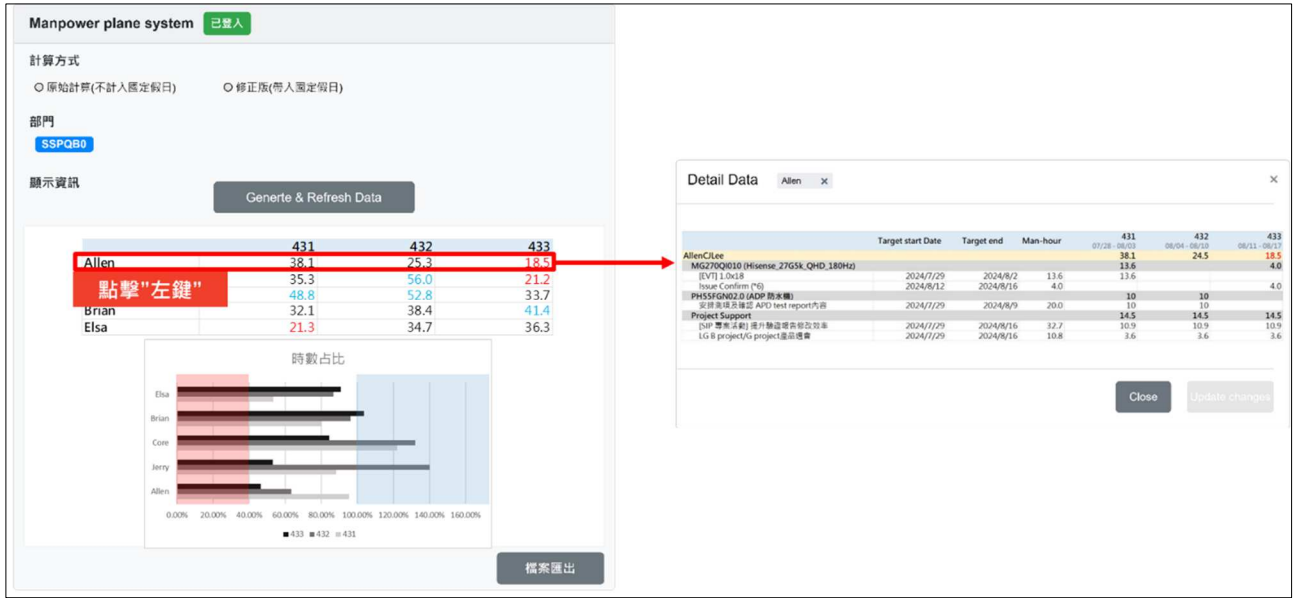
部門資料:

	431	432	433
Allen	38.1	25.3	18.5
Jerry	35.3	56.0	21.2
Core	48.8	52.8	33.7
Brian	32.1	38.4	41.4
Elsa	21.3	34.7	36.3

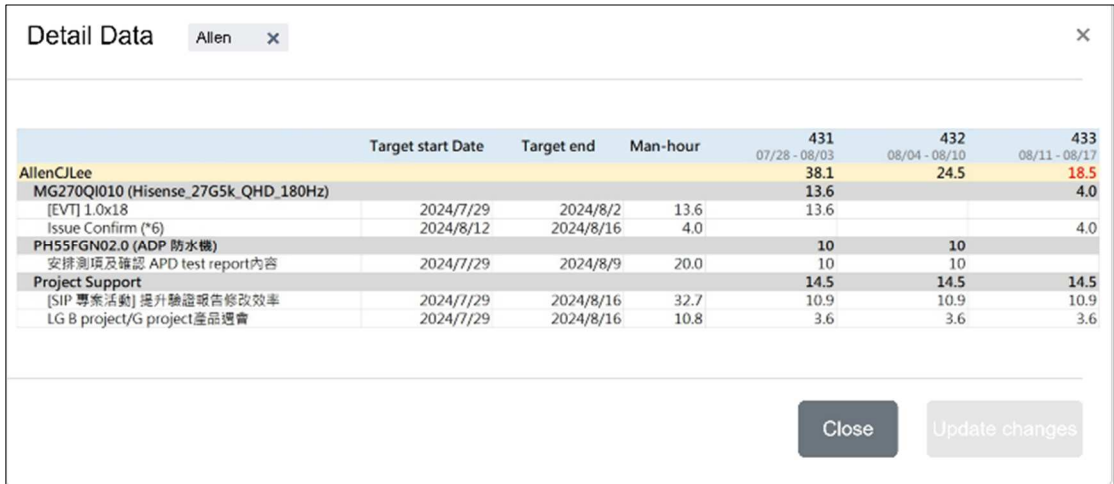


第 2.2.1 節 介面 – 詳細資料選單

顯示資訊的介面中，於內容欲觀看人員處點擊“滑鼠左鍵”，即會跳出視窗顯示該同仁的詳細資訊



顯示內容包含該同仁專案名稱、專案 task、Target Start Date、Target End、Man-hour 及當周+後 2 周資訊



- 加入多人顯示選單

若有需多人調整時數，可以於下拉選單，添加/移除 其他成員，勾選完成後詳細資料將更新為選擇選取人員資料

Detail Data

Allen x

Allen x Jerry x

	Target start Date	Target end	Man-hour	431 07/28 - 08/03	432 08/04 - 08/10	433 08/11 - 08/17
AllenC/Lee				38.1	24.5	18.5
MG270QI010 (Hisense_27G5k_QHD_180Hz)				13.6		4.0
[EVT] 1.0x18	2024/7/29	2024/8/2	13.6	13.6		
Issue Confirm (*6)	2024/8/12	2024/8/16	4.0			4.0
PH55FGN02.0 (ADP 防水機)				10	10	
安排測場及確認 APD test report內容	2024/7/29	2024/8/9	20.0	10	10	
Project Support				14.5	14.5	14.5
[SIP 專案活動] 提升驗證報告修改效率	2024/7/29	2024/8/16	32.7	10.9	10.9	10.9
LG B project/G project產品週會	2024/7/29	2024/8/16	10.8	3.6	3.6	3.6

Close Update changes

- Edit Item

直接點選 Target Start Date、Target End、Man-hour，即可調整內容

或可於該 Task 上點擊 “左鍵兩下”，即可進入到細節編輯模式，可指定要將專案轉給其他同仁，或是 copy 方式分給其他人

Edit Item

MG270QI010 (Hisense_27G5k_QHD_180Hz)

[EVT] 1.0x18

Allen

Man-hour: 20

Target start: 2024/05/13

Target end: 2024/10/13

○ Assign to

Cora

Cora

Jerry

Elsa

○ Copy to

Cora

Man-hour: 20

Target start: 2024/05/13

Target end: 2024/10/13

+

Cancel Save changes

調整完後按 “Save change”，即可回到上一步看見調整後的刷新狀態

● 同步更新 Jira

Detail Data

Allen

	Target start Date	Target end	Man-hour	431	432	433
AllenC.Lee				38.1	24.5	18.5
MG270Q010 (Hisense_27G5k_QHD_180Hz)				13.6		4.0
RTV1 LowIsa	2024/7/29	2024/8/2	13.6	13.6		
Issue Confirm (tg)	2024/8/12	2024/8/16	4.0			4.0
PH55FGN02.0 (ADP 版本欄)				10	10	
全球專案及製造 APD test report內容	2024/7/29	2024/8/9	20.0			
Project Support				14.5	14.5	14.5
[LIP 專案支援] 提升製造報告報告效率	2024/7/29	2024/8/16	12.7	10.9	10.9	10.9
LG-B project/GS project產品運費	2024/7/29	2024/8/16	10.8	3.6	3.6	3.6

Close Update changes

Detail Data

Allen

	Target start Date	Target end	Man-hour	431	432	433
AllenC.Lee				38.1	24.5	18.5
MG270Q010 (Hisense_27G5k_QHD_180Hz)				13.6		4.0
RTV1 LowIsa	2024/7/29	2024/8/2	13.6	13.6		
Issue Confirm (tg)	2024/8/12	2024/8/16	4.0			4.0
PH55FGN02.0 (ADP 版本欄)				10	10	
全球專案及製造 APD test report內容	2024/7/29	2024/8/9	20.0			
Project Support				14.5	14.5	14.5
[LIP 專案支援] 提升製造報告報告效率	2024/7/29	2024/8/16	12.7	10.9	10.9	10.9
LG-B project/GS project產品運費	2024/7/29	2024/8/16	10.8	3.6	3.6	3.6

Close Update changes

資料同步更新Jira：

- 無資料修改：
"Update Changes"反灰無作用
- 有資料修改：
"Update Changes"顯示藍色，
若按下後，則同步修改Jira上資料

第 2.2.1 節 介面 - 檔案匯出

可將完整的檔案匯出，選擇匯出路徑做存檔

Manpower plane system 已登入

計算方式
☐ 原始計算(不計入固定假日) ☐ 修正版(帶入固定假日)

部門
SSPQB0

顯示資訊
Generate & Refresh Data

	431	432	433
Allen	38.1	25.3	18.5
Jerry	35.3	56.0	21.2
Core	48.8	52.8	33.7
Brian	32.1	38.4	41.4
Elsa	21.3	34.7	36.3

時數占比

Elsa

Brian

Core

Jerry

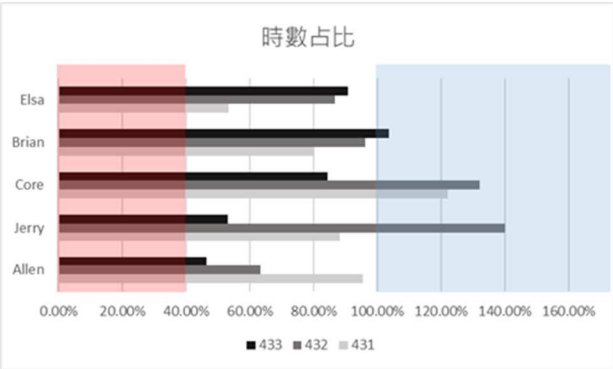
Allen

檔案匯出

檔案匯出路徑
Choose path Browse
檔案匯出

匯出內容如下，其內容包含: 匯出預覽資訊 + 個人詳細時數資料

	431	432	433
Allen	38.1	25.3	18.5
Jerry	35.3	56	21.2
Core	48.8	52.8	33.7
Brian	32.1	38.4	41.4
Elsa	21.3	34.7	36.3



列標籤	431	432	433
AllenCJLee 李嘉仁	38.1	25.3	25.3
MG270QI01.0 (Hisense_27G5K_QHD_180Hz)	13.6	0.8	0.8
[EVT] 1.0x18	13.6		
Issue Confirm (*6)		0.8	0.8
PH55FGN02.0 (ADP 防水機)	10	10	10
安排測項及確認 ADP test report 內容	10	10	10
Project Support	14.5	14.5	14.5
[SIP 專案活動] 提升驗證報告修改效率	10.9	10.9	10.9
LG B Project/G Project 產品週會	3.6	3.6	3.6
•			
•			
•			

第 3 章 MAN POWER TOOL 軟體架構簡介

根據前兩章的介紹，我們將在本章中架構一個軟體結構去完成軟體需求。在第 1 章中，我們分析了整個 Man Power Tool 系統，可以分為：

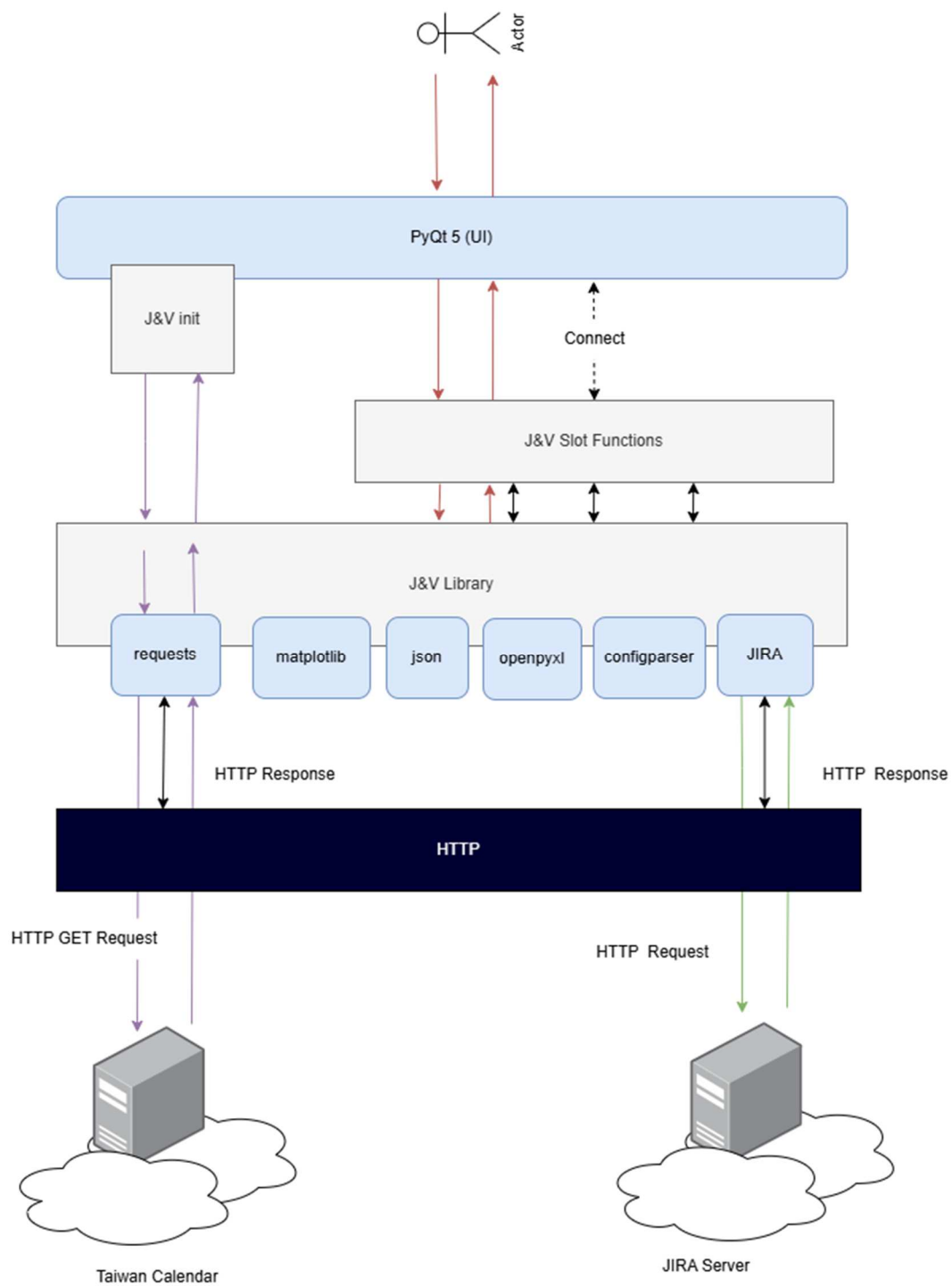
- (1) 取得工時紀錄之資料；
- (2) 處理資料；
- (3) 顯示資料；

同時我們也介紹了我們如何使用 RESTful API 去「取得工時紀錄之資料」。而在第 2 章，我們也介紹我們「處理資料」和「顯示資料」的需求。而在本章中，我們會把這些需求以具體的方式呈現出來。

第 3.1 軟體設計架構

由於空間不足，我們將 Man Power Tool 的軟體設計架構圖放置在下一頁，如下頁圖，不難發現 Man Power Tool 的軟體架構是由一些淺藍色方塊、一些淺灰白色方塊以及一個黑色方塊「HTTP」所構成；這裡，值得我們注意的只有圖中的淺藍色方塊以及淺灰白色方塊；淺藍色方塊為 Python 開放原始碼的模組，而淺灰白色方塊為我們自行封裝的模組，主要的目的是橋接使用者介面、Man Power Tool 的運作演算法，以及封裝 Python 開放原始碼模組。

首先，我們會利用 Python 的 PyQt5 套件打造 Man Power Tool 使用者介面(User Interface, UI)，然後在 UI 與 Python 開放原始碼模組之前，隔一層由我們所撰寫的程式模組去銜接這兩層。如此，Man Power Tool 的軟體架構便組建完成。更多的內容，我們會在之後的章節做更完整的介紹。



第 3.2 節 軟體套件簡介

本節延續上節，針對架構圖中淺藍色方塊的 Python 開放原始碼套件做一個簡介。

第 3.2.1 節 PYTHON PYQT5 套件

PyQt5 套件是基於 Qt 5 的 Python 開發套件；換言之，我們可以利用該套件的 Qt Designer 開發使用者介面，而後轉成 Python 程式碼。轉換後，便可以將之整合至我們的 Man Power Tool 之中。

第 3.2.2 節 PYTHON JSON 套件

由於大多數網站伺服器，特別是符合 RESTful API 的網站伺服器，伺服器回應的 HTTP Response 的資料格式大多都是 JSON 格式。因此為了方便開發，我們引入了這個 Json 套件，讓我們方便處理 JSON 字串。

第 3.2.3 節 PYTHON REQUESTS 套件

由於我們必須對網站伺服器發起 HTTP 請求，因此需要這個 Requests 套件去發送 HTTP API 或是 HTTP RESTful API 對網站伺服器進行增(Create)、查(Read)、改(Update)、刪>Delete)，即 CRUD。

第 3.2.4 節 PYTHON CONFIGPARSER 套件

關於程式的一些預設值，或是設定值，為了能夠實現更佳彈性的設計，我們通常會把這些設定值獨立於程式之外，成為設定檔案。最常見的設定檔案便為 ini 設定檔案，而 configparser 套件便是用來剖析這類設定檔案的套件。

第 3.2.5 節 PYTHON JIRA 套件

除了使用 Requests 套件可以呼叫 HTTP API 以及 HTTP RESTful API，針對 JIRA 的伺服器，Python 也有專門的 jira 套件可以對 JIRA 伺服器進行操作。

第 3.2.6 節 PYTHON OPENPYXL 套件

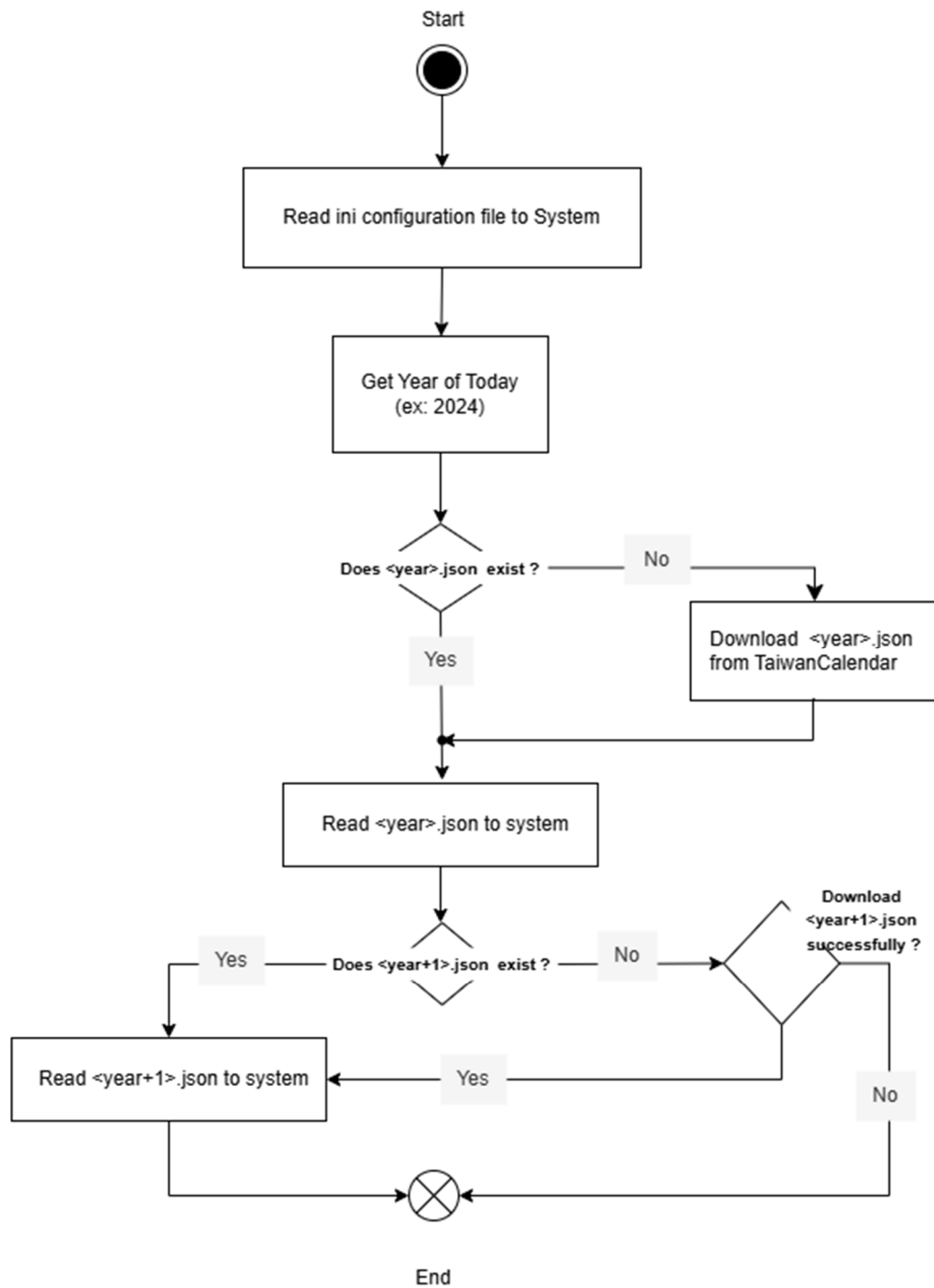
除了取得以及處理資料外，Man Power Tool 還需要將其整理成 EXCEL，openpyxl 便是方便我們操作 EXCEL 的套件。

第 3.2.7 節 PYTHON MATPLOGLIB 套件

Matplotlib 是 Python 第三方函式庫中相當重要且廣泛被使用的資料視覺化函式庫，簡單理解就是幫助開發者將數據資料以圖表的方式呈現的函式庫，在本次實作中，我們將會把它與 PyQt5 串聯與開發。

第 3.3 節 MAN POWER TOOL 初始化執行流程

這一節將介紹 Man Power Tool 初始化執行流程；當 Man Power Tool 程式被使用者點擊後，程式便會開始運作，而在運作一開始便會執行初始化之動作。如下圖：



首先程式一開始會在指定目錄下讀取 ini 設定檔案，並且把這些設定從檔案載入至記憶體；而後系統需要取得今年台灣的行事曆，該行事曆可以至如下網址下載：

<https://github.com/ruyut/TaiwanCalendar>

因此我們必須取得今天日期的年份，並且根據年份至該網站取得以年份命名的 JSON 檔案，以今年 2024 年為例，便需要取得 2024.json 檔案。但問題來了，這個檔案理論上應該只需要下載一次便可，而不需要每次 Man Power Tool 被點擊都被執行一次才對；於是我們在取得今年的年份後，便可以至指定的位置去判斷 <year>.json 的檔案是否存在？

如果該檔案不存在，便要至 TaiwanCalendar 網站下載 <year>.json 檔案；而如果檔案存在，那便讀取 <year>.json 檔案，並且選擇一個適當的資料構將之載入記憶體，以便後續程式演算法的使用與呈現。

由於可能存在某週會有跨年度的可能，因此我們仍然需要事先去抓取下一年度的行事曆，所以在完成讀取今年年度之行事曆的 json 檔案後，我們仍需要嘗試去 TaiwanCalendar 網站獲取下年度的 <year+1>.json 的資料；然而這個動作是有可能不成功的，因為下年度的行事曆很可能尚未公布，因此若已經存在 <year+1>.json 或成功從 TaiwanCalendar 網站獲得 <year+1>.json，則將之載入至記憶體中。若仍然無法獲得 <year+1>.json 則等待下次使用者開啟 Man Power Tool 時再嘗試下載取得 <year+1>.json。