

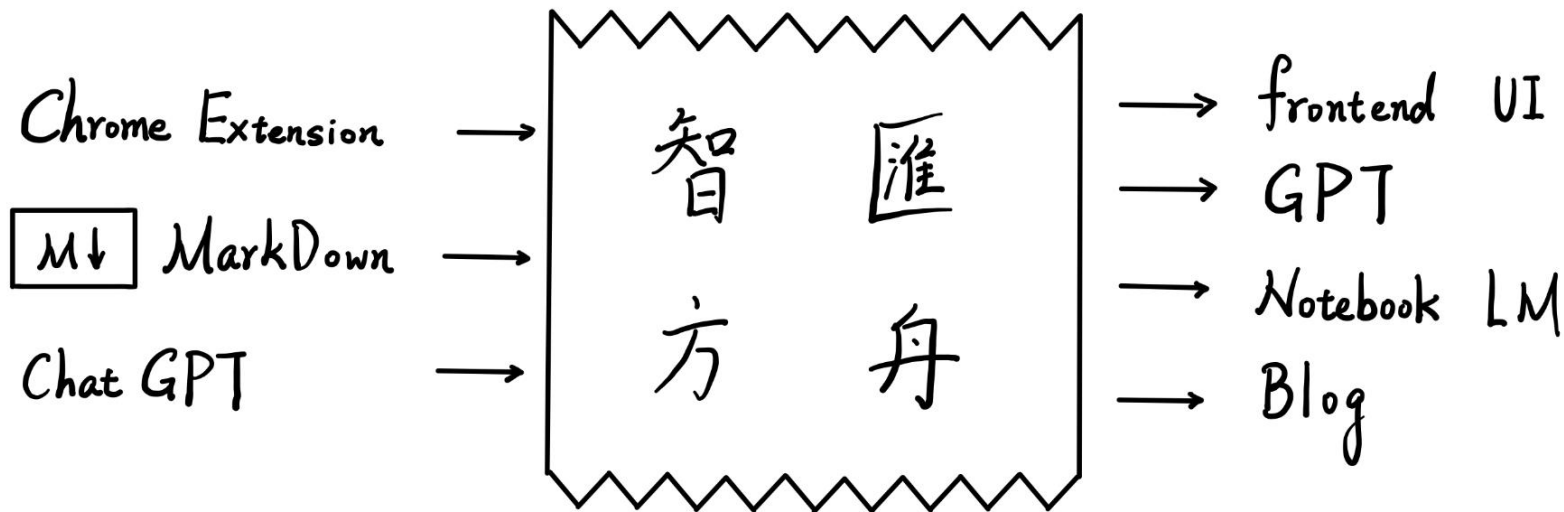
智匯方舟

台大資工 劉冠甫
台大物理 魏祺誠
交大資工 郭勝威

我們的專案



重要資料的諾亞方舟



01 背景

AI時代中重要的能力
親身經驗
我們想要解決的問題
困難點

02 同類產品探討

Save to Notion + API
Supermemory
我們的解法

03 專案介紹

總覽
產品介紹
程式架構
架構優勢

04 demo&未來的展望

Demo
未來展望

05 結論&QA

總結
我們學到了什麼
QA

01

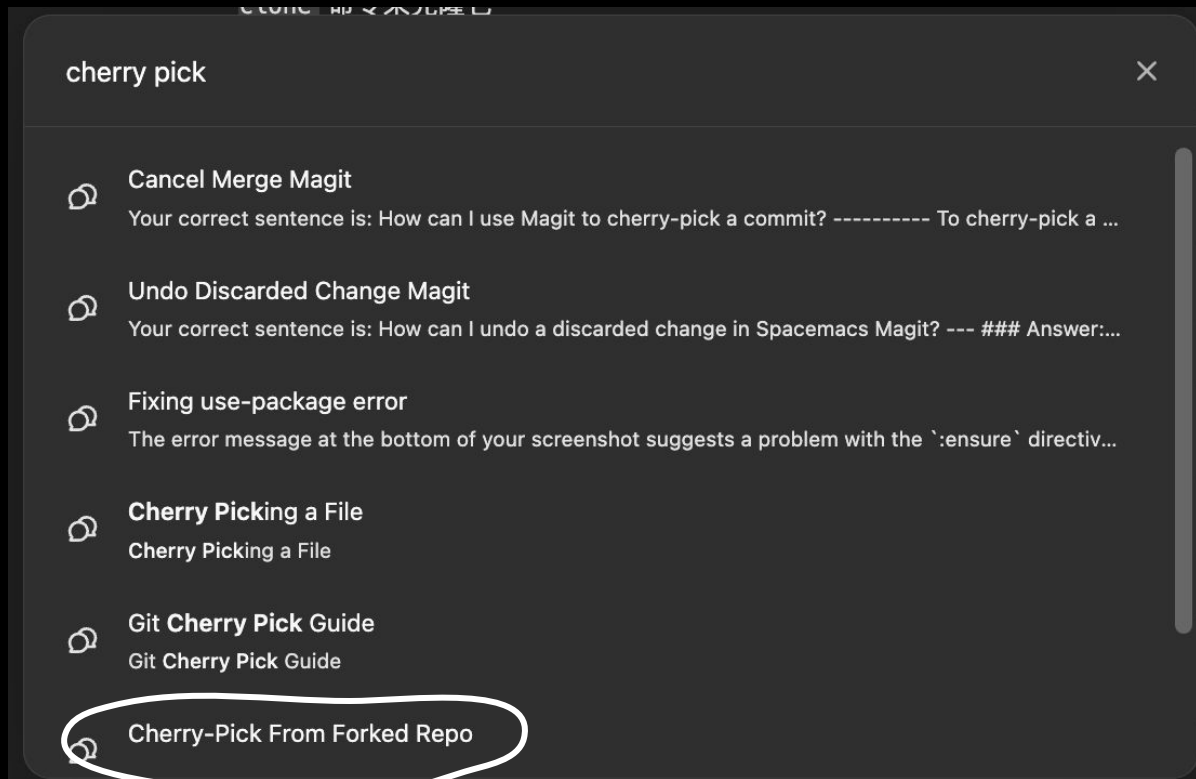
背景

親身經驗

問題: how to cherry-pick from
other repo?

第一步: Chatgpt

我記得自己這
個禮拜有問過
一樣的問題?



看完了長長的對話紀錄： 不對啊，我記得我看過更簡單 的做法

那麼怎麼 cherry-pick 不同 repo (同一個專案 fork 出來的)

- ⦿ 要在 Git 中從一個 fork 的 repository cherry-pick 提交到另一個，你需要先將那個 fork 加入到你本地的 repository 作為一個遠程參考，然後才能 cherry-pick 提交。以下是一個基本的步驟指南：

1. 添加遠程參考：首先，確保你有那個 repository 的一個本地副本。如果沒有，你可以使用 `git clone` 來克隆它。

```
bash
git clone -> 添加 repo URL
cd <克隆下來的目錄名>
```

然後，添加 fork repository 作為一個新的遠程參考 (通常稱為 remote)：

```
bash
git remote add <fork 名稱> <fork repo URL>
```

例如，如果 fork 名稱是 "forked_repo"，你可以這樣做：

```
bash
git remote add forked_repo <fork repo URL>
```

2. 獲取遠程變更：使用 `git fetch` 命令從新加的遠程參考中獲取變更。

```
bash
git fetch <fork 名稱>
```

這會下載 fork repository 的所有分支和提交，但不會合併到你的本地分支。

3. Cherry-pick 提交：現在，你可以使用 `git cherry-pick` 命令來選擇要合併的提交。首先，確保你處於你要選擇提交的分支上。如果無聊，可以使用 `git checkout` 切換分支。

然後使用以下命令進行 cherry-pick

```
bash
git cherry-pick <提交的 SHA-1 標籤ID>
```

這裡的 <提交的 SHA-1 標籤ID> 是你想複製 fork 中 cherry-pick 的提交的 ID。

4. 處理衝突：如果 cherry-pick 過程中出現衝突，你需要手動解決這些衝突。在解決衝突後，使用 `git add` 標記解決衝突的文件，然後使用 `git cherry-pick --continue` 完成 cherry-pick 過程。

5. 推送更改：完成這些操作後，你可以選擇將更改推送到你選擇 repository。

```
bash
git push origin <分支名稱>
```

此外，cherry-picking 是一種選擇性複製提交到另一個分支和標籤到另一個分支的方法，它不會自動解決衝突或創建分支。如果你需要經常複製提交的完整歷史記錄，可能更適合使用其他方法，如併入 (merge) 或 rebase。

那麼怎麼 diff 一個 commit

- ⦿ 查看一個 commit 的差異 (diff)，你可以使用 Git 的 `git diff` 命令。這個命令允許你查看特定提交與其他提交之間的差異。以下有幾種基本的用法：

1. 查看最新一次提交的差異：如果你只想查看最近一次提交所做的更改，可以使用：

```
bash
git diff HEAD^ HEAD
```

這裏，`HEAD` 指的是當前分支的最後提交，而 `HEAD^` 是指前一個提交的提交。

2. 查看任意提交之間的差異：如果你想查看兩個提交之間的差異，你可以使用 `git diff` 命令，然後使用以下命令：

```
bash
git diff <提交的 SHA-1 標籤ID> <提交的 SHA-1 標籤ID>
```

例如，如果提交的標籤是 `abc123`，那么命令就是：

```
bash
git diff abc123^ abc123
```

3. 查看兩個不同提交之間的差異：你也可以比較任意兩個提交之間的差異，例如，比較提交 `abc123` 和 `def456`：

```
bash
git diff abc123 def456
```

4. 查看特定文件的差異：如果你只对特定文件的更改感興趣，可以在 `git diff` 命令後面加文件名：

```
bash
git diff <提交的 SHA-1 標籤ID> <提交的 SHA-1 標籤ID> -- <文件名>
```

例如，查看提交 `abc123` 中 `example.txt` 文件的差異：

```
bash
git diff abc123^ abc123 -- example.txt
```

此外，Git 的 `diff` 標籤是一個長字符串，但通常只需要提供前幾個字 (通常是前六個)，只要它們在目標 repository 中是唯一的。使用這種方法，你可以快速查看不同提交之間的差異，而無需填寫完整的標籤和提交。

第二步：詢問ChatGPT

詢問了多次，但每次回答的都不一樣，
不知道那一個才是正確的

第三步: Google

看了三四個網頁終於找到了

TL;DR

```
git format-patch -k --stdout ${commit_hash_1}..${commit_hash_2} | git am -3 -k
```

來來回回花了快二十分鐘
而且是這個月第三次了....

我們想解決的問題

1. 紀錄下自己的解決方案
 - 不要讓努力白費
2. 解決方案可以在不同的**AI**被重複使用
 - 不然換一個產品就又要重新累積了

困難點

1. 結構性: 找到資料所費的時間

- 人工找也可以快速找到
- 提供**AI**足夠準確的資料

2. 拓展性:

- 對於新展品可以快速的接入服務

02

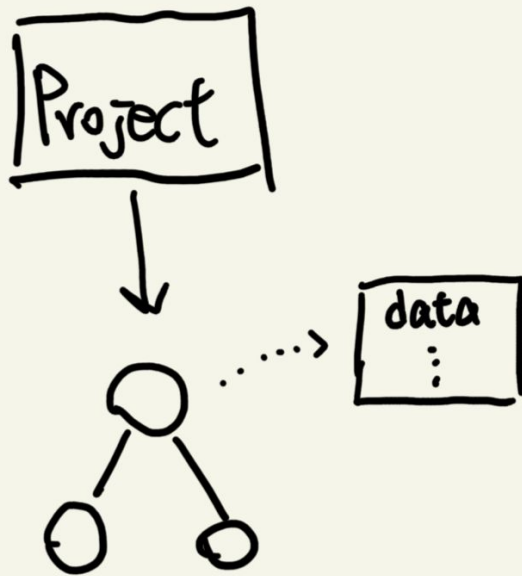
其他人 **vs.** 我們

比較

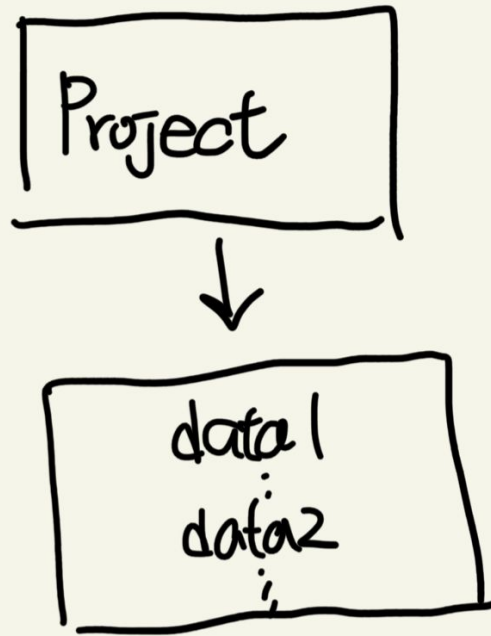
	Our	Save to Notion	Supermemory
結構性	$O(\log N)$ (Tree)	$O(N)$ (List)	$O(N)$ (List)
拓展性	佳: 可獲取多個 網頁	一般: 單一個網頁 為單位	不好: 只有AI的回答

結構性

Our







Notion/Supermemory




Notion / Supermemory

git

Table +

Aa 名稱	URL
 What is Git Atlassian Git Tutorial	atlassian.com/git...is-git
 Git - What is Git?	git-scm.com/boo...Git%3F
 Is it possible to cherry-pick a commit from another git repository?	stackoverflow.com/que...sitory
 How to cherry-pick a commit from another repository	dev.to/jma...y-4pf1

Git  Can Edit • Private Space

All Memories Web pages Tweets Documents Notes

Select Items


+

Add to your second brain.

Add a link, a note, a document, tweet, etc.


IT 邦幫忙:一起幫忙解決難題,拯救 IT 人的一天

Saved from ithelp.ithome.com.tw

ithelp.ithome.com.tw 


How to copy commits from one branch to another?

Saved from stackoverflow.com

stackoverflow.com 

【狀況題】如果你只想要某個分支的某幾個 Commit? - 為你自己學 Git | 高見龍

Saved from gitbook.tw

gitbook.tw 

Our

Root

Cherry Pick

From Another
Repository

From Another Repository

1. Using cherry-pick

```
git remote add other_repo <repo_url>
git fetch other_repo
git cherry-pick <commit_hash>
```

2. Using format-patch and am

```
git format-patch -1 <commit_hash> --stdout > commit.patch
git am < commit.patch
```

3. Using --git-dir

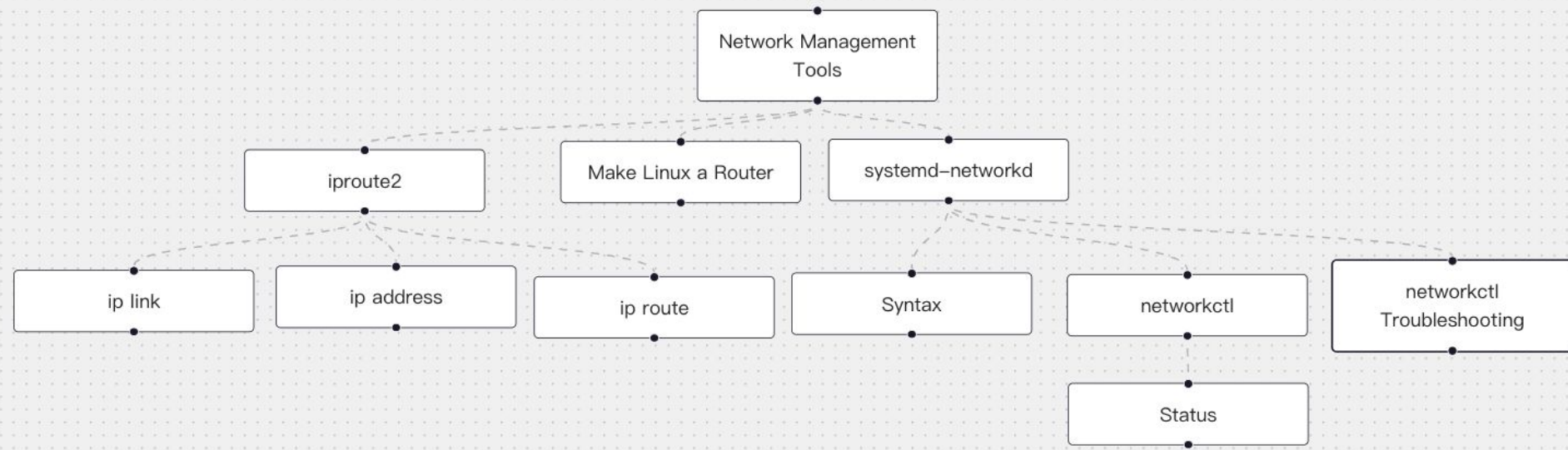
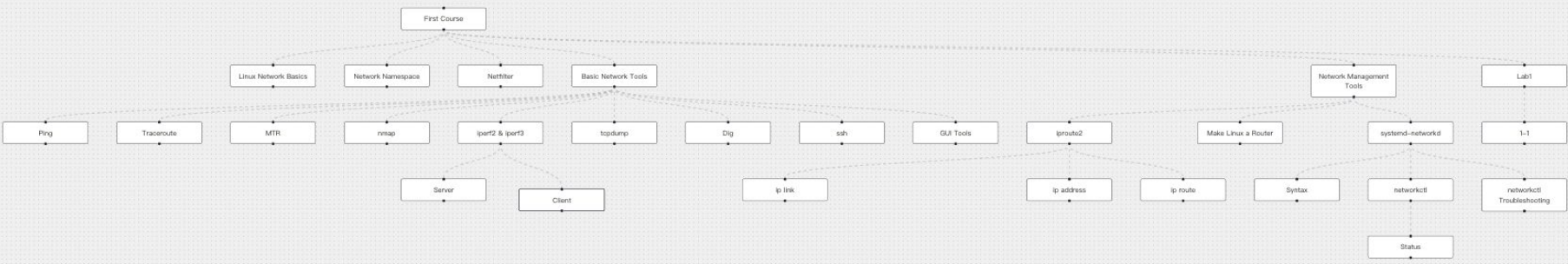
```
git --git-dir=/path/to/source_repo/.git format-patch -1 <commit_hash> --stdout | git am
```

目標用戶

1. 網路管理員/系統管理員
2. 正在學習一個複雜知識的入
- 3.

Or

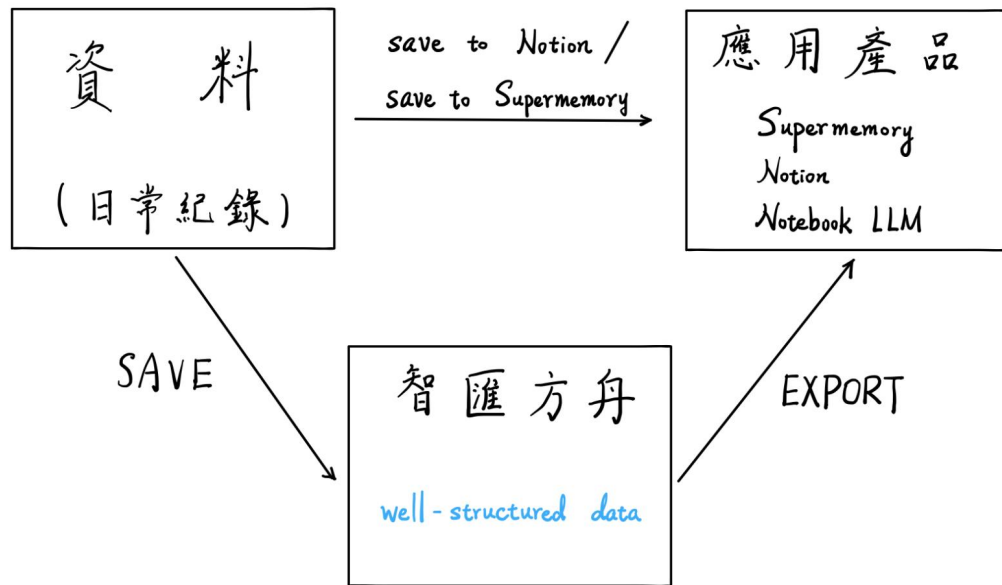
4. 在修**NASA**的資工系學生...



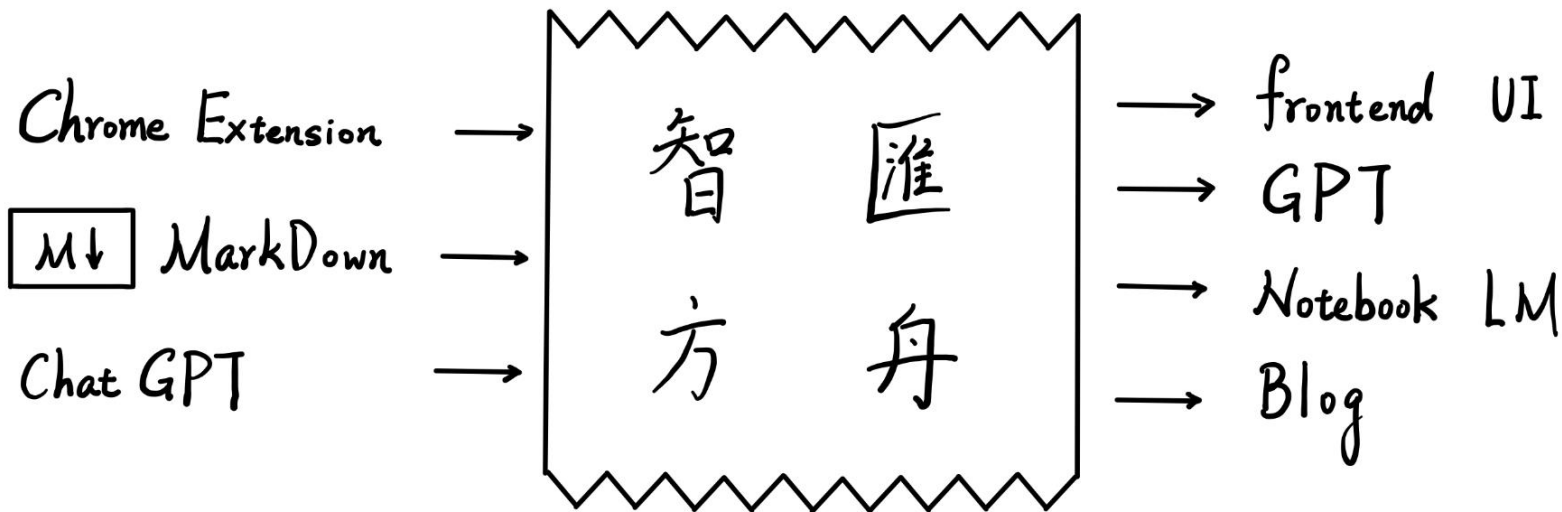
03

專案介紹

總覽



產品介紹



產品架構

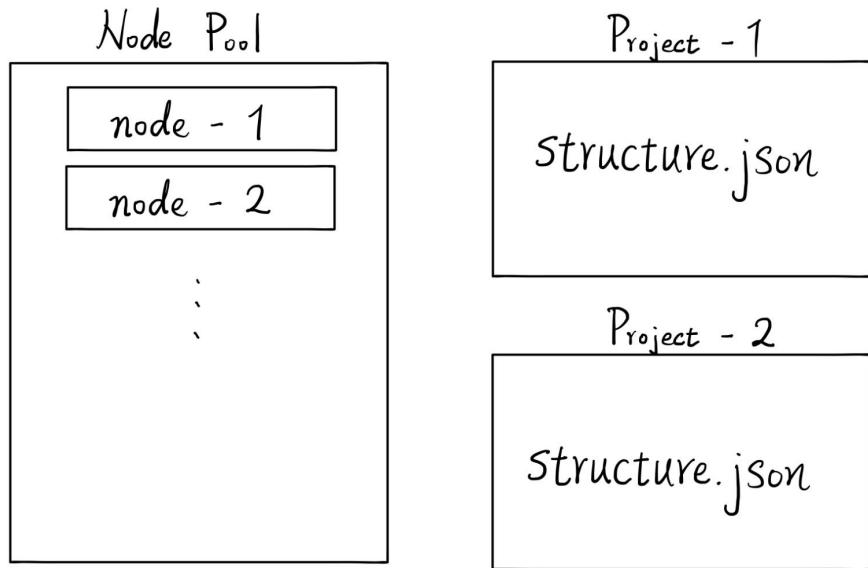
1. Node - 資料

- 一個知識單元
- 資料分Important, relate, other

2. Project - 結構

- 專門負責node間的連接

產品架構



產品架構

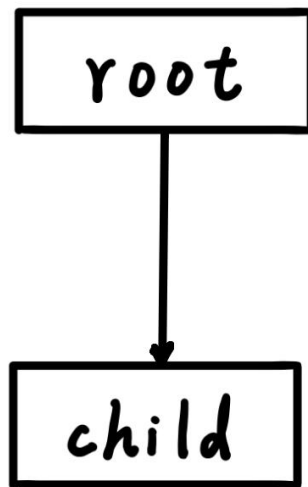
Node

```
{  
  nodeID: "  ",  
  Title: "  ",  
  important_data: [...],  
  relate_data: [...],  
  other_data: [...]  
}
```

Project

structure.json

```
{  
  nodeA: [nodeB],  
  nodeB: [],  
  nodeTitle: {  
    nodeA: root,  
    nodeB: child  
  }  
}
```



架構優勢

1. Node 內分類

- 更有彈性

2. NodePool

- 不同的**project** 可以共用一個**node**
- 好儲存

實際例子

Summarizer(chrome extension)

- 用**AI**總結當前網頁並加入**database**內

創新: auto upload

- 自動將設定好的網站加入到**relate data (e.g. stackoverflow)**

04

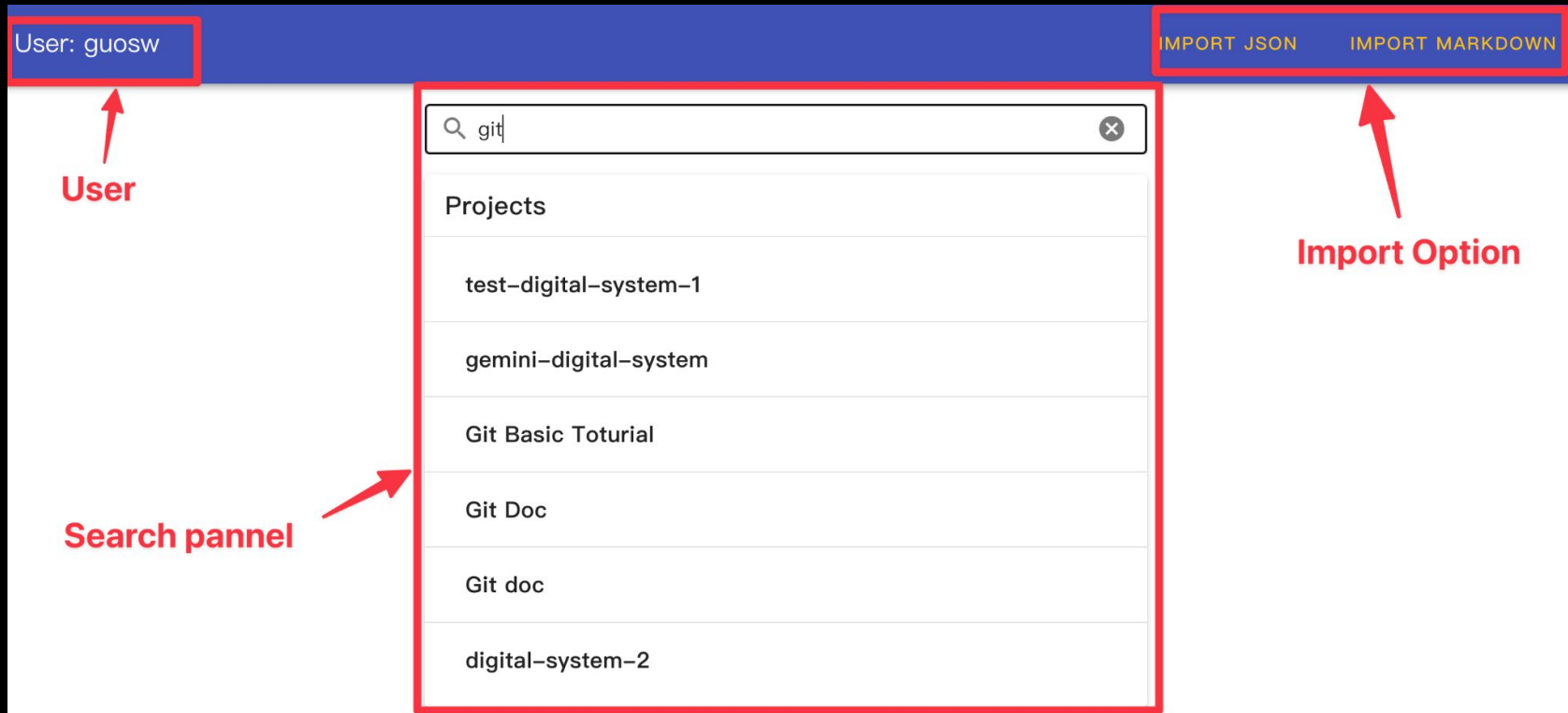
Demo & 未來展望

Demo: Git 學習

產品介紹



Frontend Introduce



Frontend Introduce

Root Node Explorer

Search Root Nodes

← 返回上層

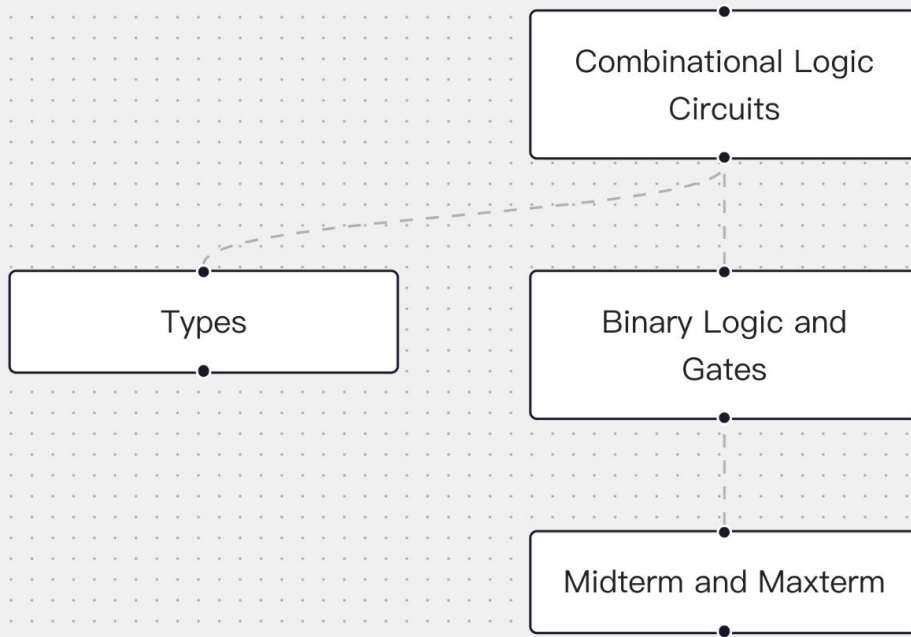
Types

Binary Logic and Gates

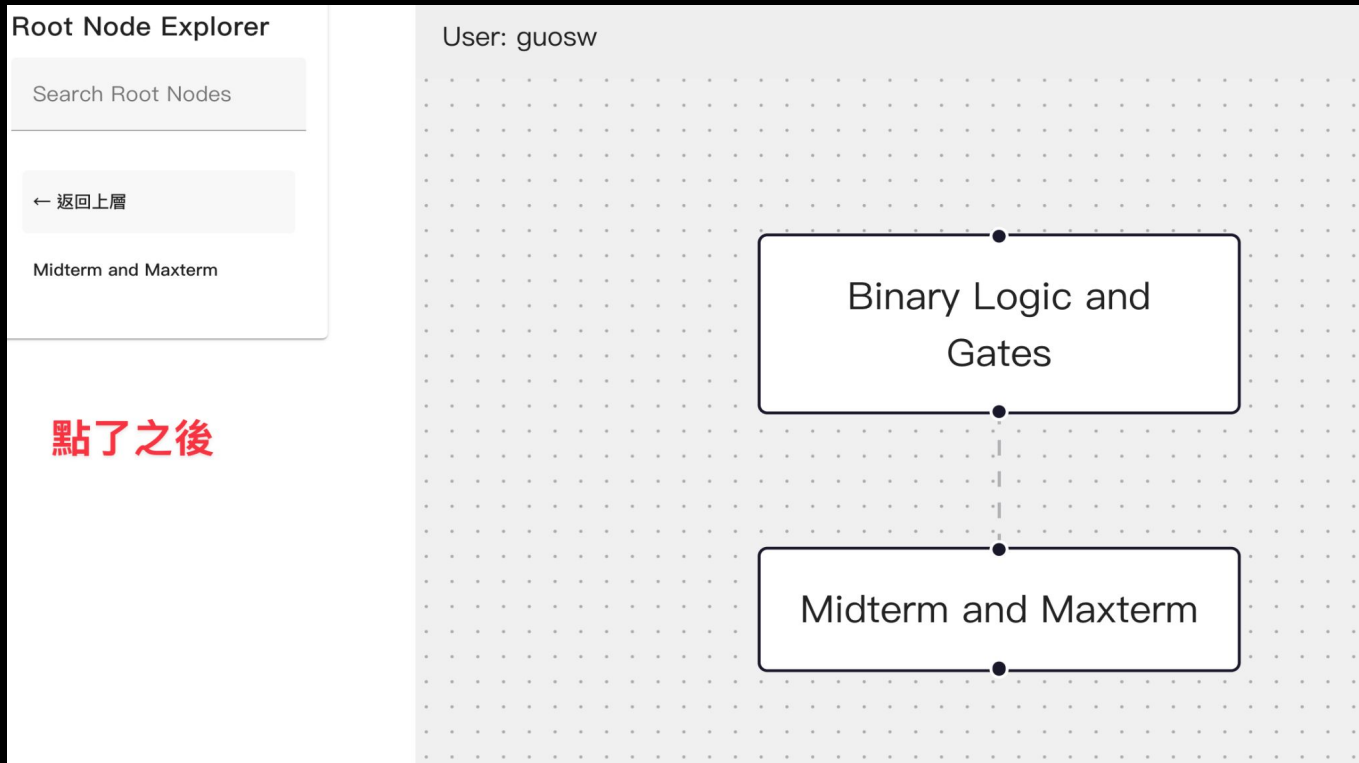
子節點
- 可點

User: guosw

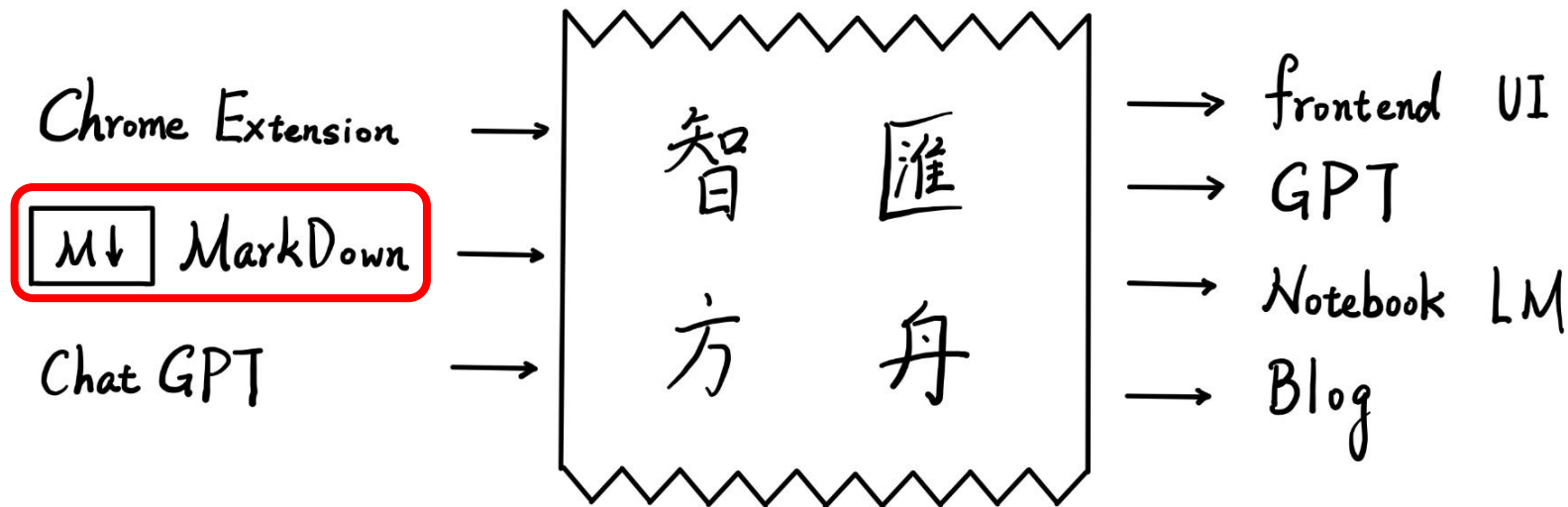
JSON STRUCTURE



Frontend Introduce



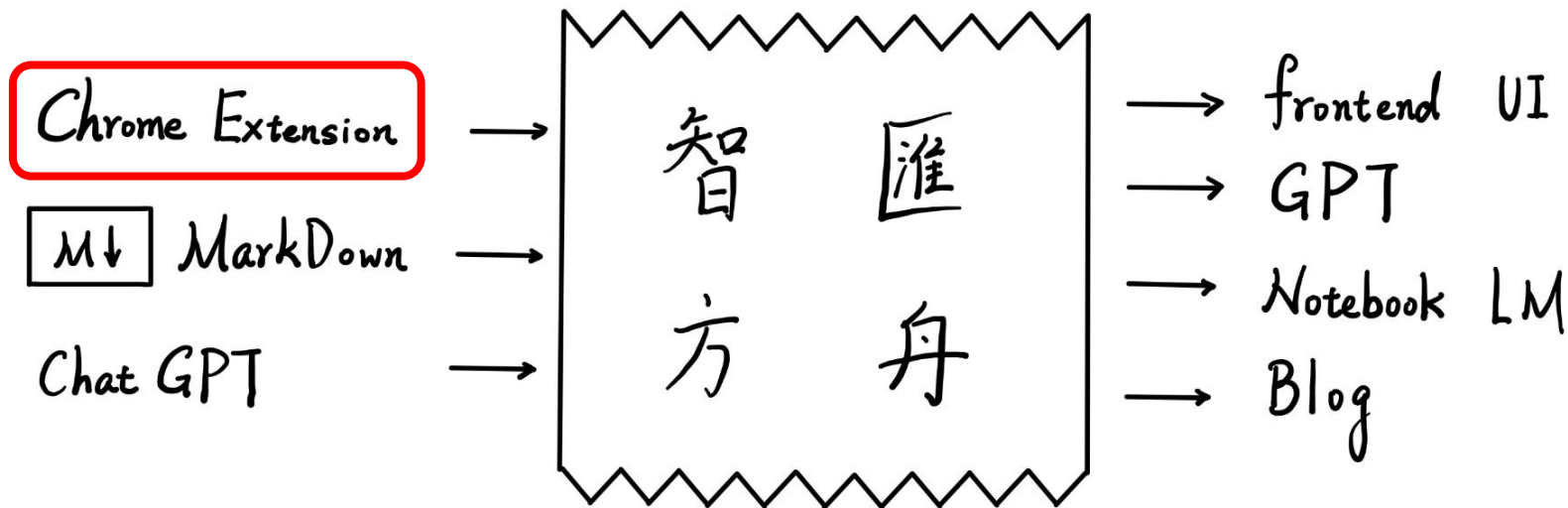
產品介紹



Import Markdown

錄影待補

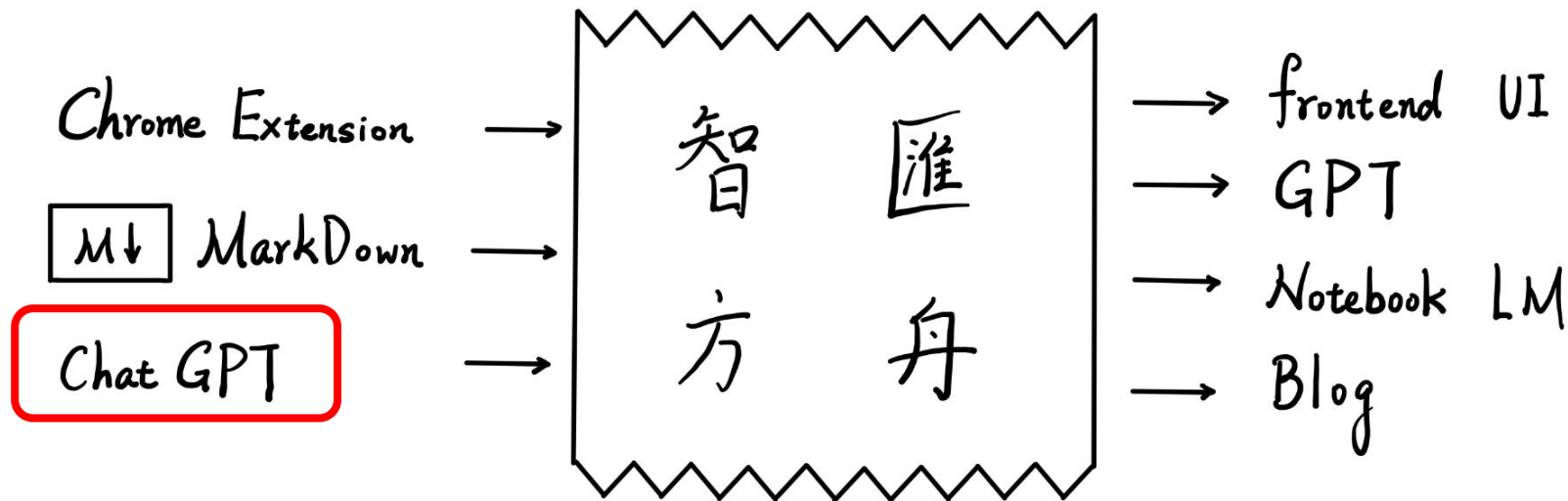
產品介紹



Summarizer

錄影待補

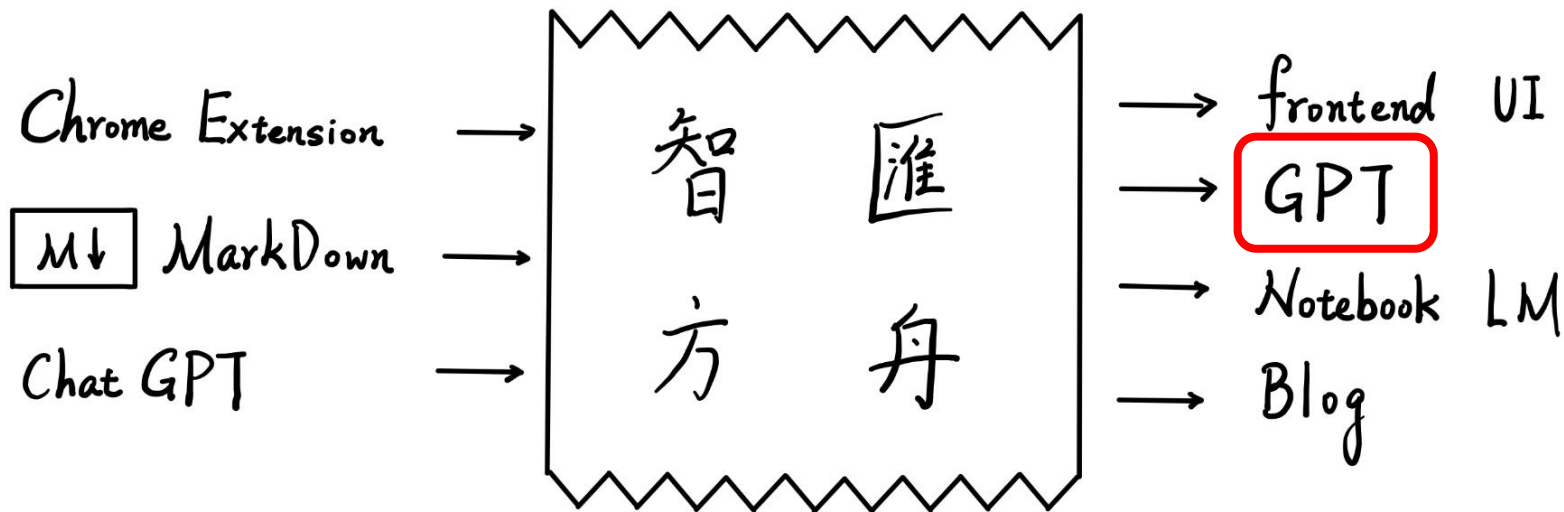
產品介紹



Chatgpt Save

錄影待補

產品介紹



Chatgpt Ask

錄影待補

未來展望

API Server

1. 可以存文檔
2. 增加對node內文的搜索

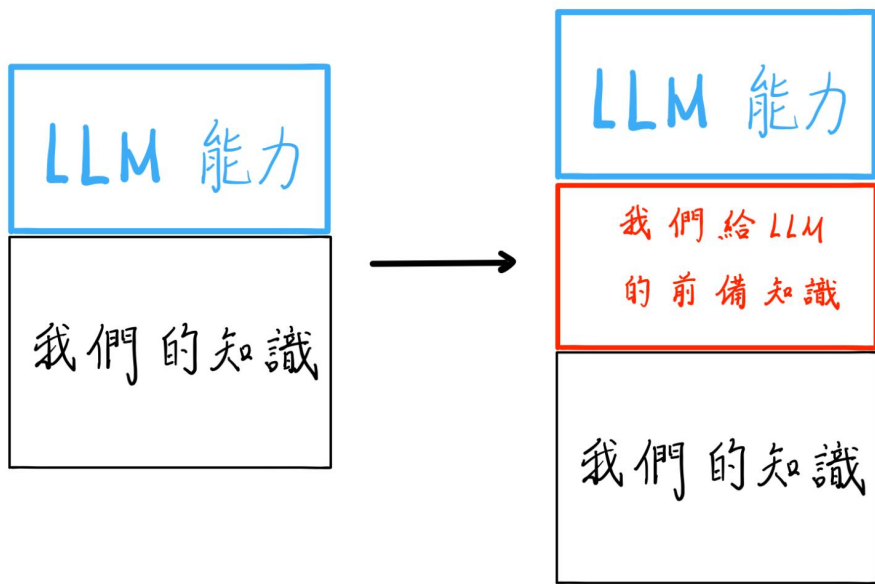
插件

- 自動生成個人的技術Blog

05

結論

總結



我們學到什麼

Summarizer

- javascript

前端顯示界面

- Vue, Vue flow

API Server

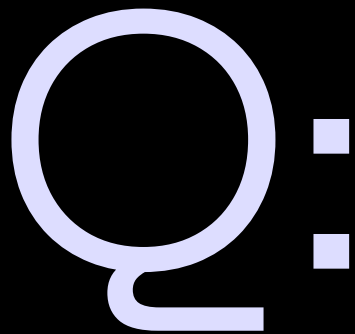
- fastapi

我們學到什麼

1. 如何快速迭代出可用的版本
2. 順著一開始的產品想法不斷擴大目標

QA Time

THANK YOU



在llm發展的時代, 重要的能力是什麼

A:

1. 提出問題

- 前知識很重要(決定問題的品質)

2. 存下答案

- 避免做白工
- 累積前知識

I want to use cherry pick from another repository
<repo_url>

1. please give me the one line code
2. please use git format-patch and git am
3. don't use git fetch

🔍 To apply a commit from another repository without using `git fetch`, you can create a patch from the desired commit in the source repository and then apply it to your current repository. Here's how you can do it:

1. One-line command:

bash

📋 複製 ✎ 編輯

```
ce_repo>/.git format-patch -k -1 --stdout <commit_hash> | git am -3 -k
```



Save to notion+API

Save to Notion

<

form

網路知識庫

⚙️

Aa Name

PATH 雲端課程

≡ tag

數位學習資源

≡ 喜歡的原因

可以跟OHA雲端教室結合，也可以單獨使用

3 more fields

Clip Page

▼

雲端知識庫

Table List + Filter Sort 🔍 ... New ▼

Aa Name	Tags	🔗 URL	+ ...
 Webduino模擬器「瘋」神榜：用Web:Bit陣列學Youtube影片控制與玩實果遊戲	IoT	https://harmonica80.blogspot.com/2022/12/webduinowe-bbityoutube.html	
 Sweet Home 3D 最 Easy 開講：Monster Mash 輕鬆將2D照片轉成3D模型，歡喜入住室內設計空間！	3D	https://harmonica80.blogspot.com/2023/01/sweet-home-3d-easy-monster-mash-2d3d.html	

使用的工具

1. 查資料: perplexity
2. 問問題: Chatgpt
3. 學習: NotebookLm
4. 程式錯誤: Google stackoverflow

第二步：詢問GPT

跟一開始的一樣

我要怎麼從其他repo cherry pick

- ▶ 你的正確句子是 =>
我要如何從其他的 repo 進行 cherry-pick ?

從其他 repo 進行 cherry-pick 的方法

如果你想從另一個 repository (repo) 中挑選特定的 commit，通常可以透過以下步驟來完成：

方法 1：使用 `git fetch + cherry-pick`

1. 切換到目標分支（確保你在正確的分支上進行 cherry-pick）：

```
bash
```

複製 編輯

給了很多做法，但不知道用那一個

有沒有更好的做法

- ▶ 你的正確句子是 =>
有沒有更好的做法？

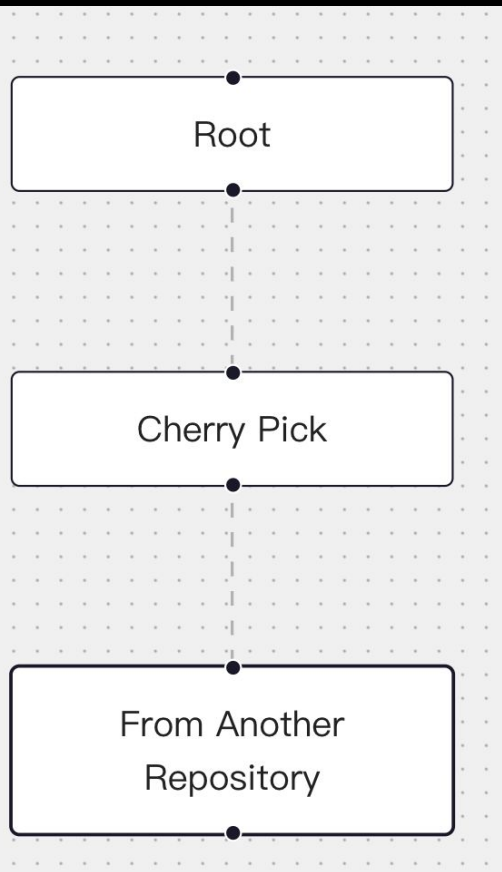
更好的方法來從其他 repo 拿 commit

`git cherry-pick` 雖然有效，但如果你經常從另一個 repo 引入變更，可能會有更好的方式來處理，例如使用 `git merge` 或 `git rebase`，這樣可以避免手動挑選 commit，並且保持更完整的 commit 歷史。

以下是一些更好的做法，視你的需求選擇合適的方法：

方法 1：使用 `git format-patch + git am`（適合從別 repo 搬運特定 commit）

知識架構 - 即使資料很多也一目了然



From Another Repository

1. Using cherry-pick

```
1 git remote add other_repo <repo_url>
2 git fetch other_repo
3 git cherry-pick <commit_hash>
```

2. Using format-patch and am

```
1 git format-patch -1 <commit_hash> --stdout > commit.patch
2 git am < commit.patch
```

3. Using --git-dir

```
1 git --git-dir=/path/to/source_repo/.git format-patch -1 <commit_hash> --stdout | git am
```

問完之後直接 呼叫API

```
user: guosw  
urltype: important  
/auto_sum
```

▶ 已與 api.driseam.com 交談 ▾

已與 api.driseam.com 交談 ▾

已與 api.driseam.com 交談 ▾

已與 api.driseam.com 交談 ▾

已與 api.driseam.com 交談 ▾

已與 api.driseam.com 交談 ▾

已與 api.driseam.com 交談 ▾

The information about cherry-picking from another repository has been successfully stored in your project "Git" under the node "From Another Repository."

Let me know if you need any modifications or additional details! 🚀

user: guosw
/answer
how can I cherry pick from other repository

已與 api.driseam.com 交談 ▾

已與 api.driseam.com 交談 ▾

已與 api.driseam.com 交談 ▾

已與 api.driseam.com 交談 ▾

To cherry-pick a commit from another repository, you have a few options:

1. Using `cherry-pick`

```
bash
git remote add other_repo <repo_url>
git fetch other_repo
git cherry-pick <commit_hash>
```

複製 編輯

2. Using `format-patch` and `am`

```
bash
git format-patch -1 <commit_hash> --stdout > commit.patch
git am < commit.patch>
```

複製 編輯

3. Using `--git-dir`

```
bash
git --git-dir=/path/to/source_repo/.git format-patch -1 <commit_hash> --stdout | git am
```

複製 編輯

Let me know if you need further clarification! 🚀

下次再問的時候就可以快速找到檔案

大綱

1. Out: FrontEnd introduction
2. In: Import markdown
3. In: summarizer(chrome extension)
4. In: ChatGPT save conversation
5. Out: ChatGPT ask conversation

Save to notion+API

1. 精確性 🤔:

- 可以幫資料庫加上**Tag**
- 但資料之間沒辦法有相關性

2. 傳輸性 👍:

- 有提供**API**可以分配

3. 拓展性 👎:

- 無法處理資料之間關聯

Supermemory

1. 精確性 👍:
 - 直接透過LLM+Rag回答
2. 傳輸性 🤔:
 - 只提供詢問LLM的回答
3. 拓展性 🤔:
 - 無法處理資料之間關聯
 - 分類要自己手動搬

Supermemory



我們的解法

1. api server

- 寫插件調用api即可

2. 良好的架構設計, Project + Node

- 易傳輸
- 高拓展性
- 精確性