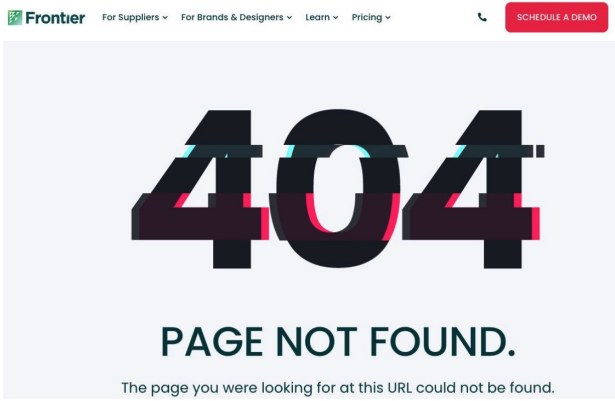
System Design 101

Ting 2024/03/13

Outline

1. HTTP Status Codes
2. What is a webhook?
3. Code First vs. API First
4. REST API vs. GraphQL

WHY Need HTTP Status Codes?



1. 改善**網站使用者**體驗

   用戶輸入了無效網址時，可以讓用戶明確地知道

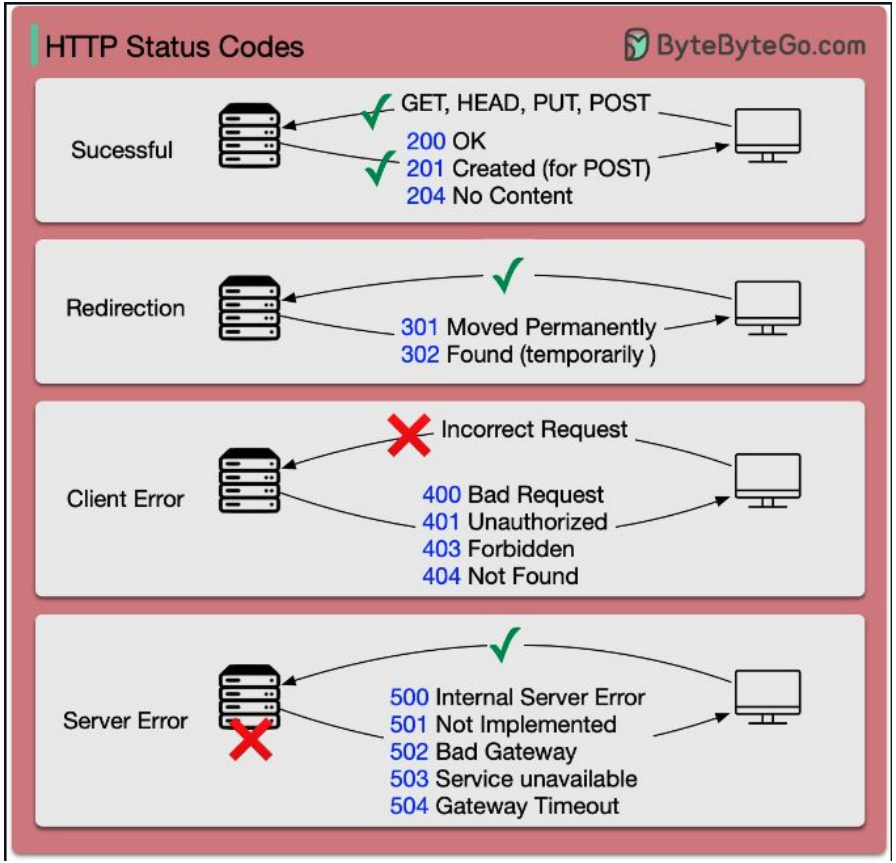
   該網址資源不存在，而不是獲得一個模糊的錯誤訊息。(404 Not Found)

2. 讓**開發人員**更易處理錯誤和異常情況

   開發人員可以根據不同的情況進行適當的錯誤處理和故障排除。(500 Internal Server Error表示伺服器遇到內部錯誤，401 Unauthorized表示需要身份驗證但未提供有效認證資訊。)

3. 提供溝通和回饋

   HTTP狀態碼是**伺服器(Server side)對於客戶端(Client side)請求的回應**，提供一種標準化的方式來溝通伺服器的處理結果。客戶端可以透過狀態碼得知伺服器的回應，並根據不同的狀態碼進行相應的處理。



https://github.com/ByteByteGoHq/system-design-101/blob/main/images/http-status-code.jpg

2 - Successful

200 OK

請求成功，伺服器正確回應請求

3 - Redirection

302 Found：當訪問一個網址，但該網址已被重定向到其他網址

4 - Client Error

401 Unauthorized：試圖訪問需要登錄的頁面，但未提供正確的登錄資訊

404 Not Found：請求的資源在伺服器上找不到

5 - Server Error

502 Bad Gateway 當用戶端向伺服器發送請求時，中介者伺服器無法從上游伺服器（通常是網頁伺服器）獲得有效的回應

WHY Need Webhook?

讓**不同的應用程式之間** 實現**即時的資訊交換** 和**數據同步**

- **即時通知**:

  當某個事件發生，Webhook可立即通知接收方，而不需要等待定期輪詢（ polling API）或等待回應。

- **自動化流程**:

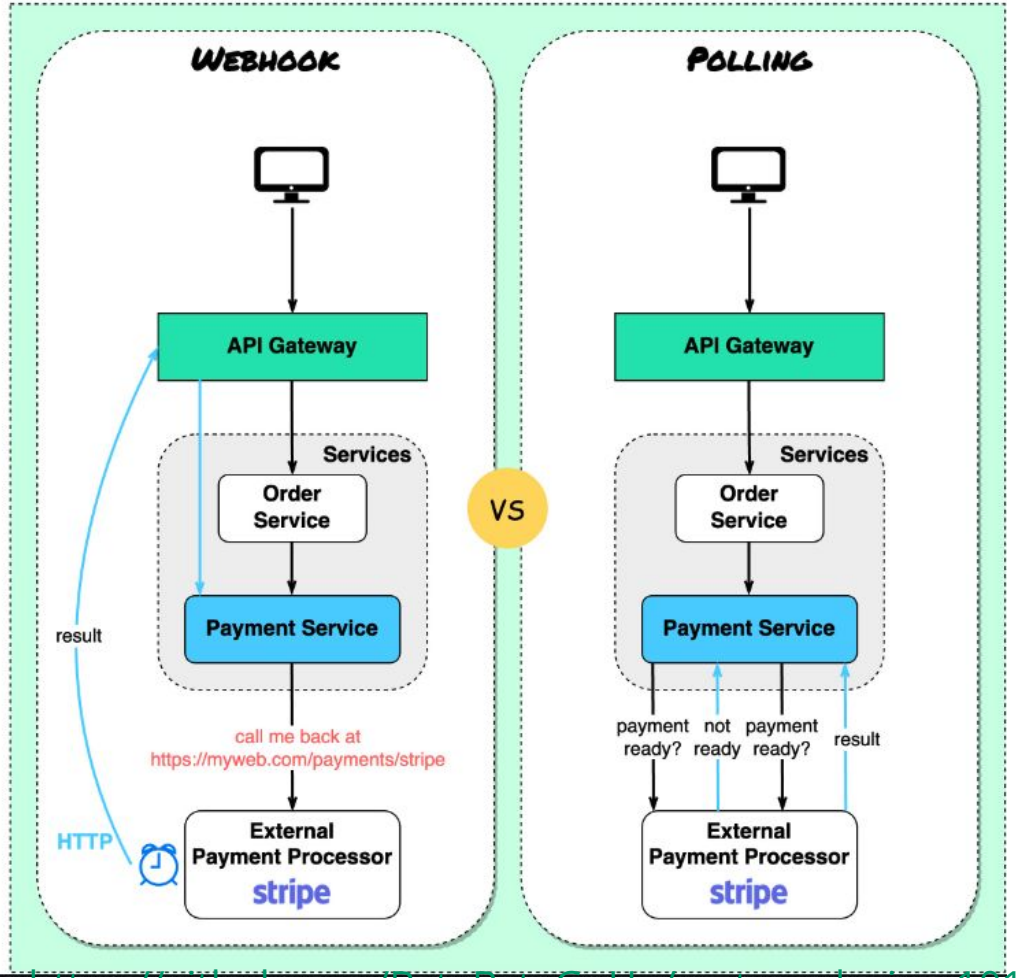
  Webhook觸發自動化流程，例如在提交表單後自動發送電子郵件，或更改資料庫中的 內容。

- **資料同步**:

  Webhook可用於將數據從一個應用程式同步到另一個應用程式，例如將電子商務網站的訂單同步到庫存管理系統中。

- **數據分析**:

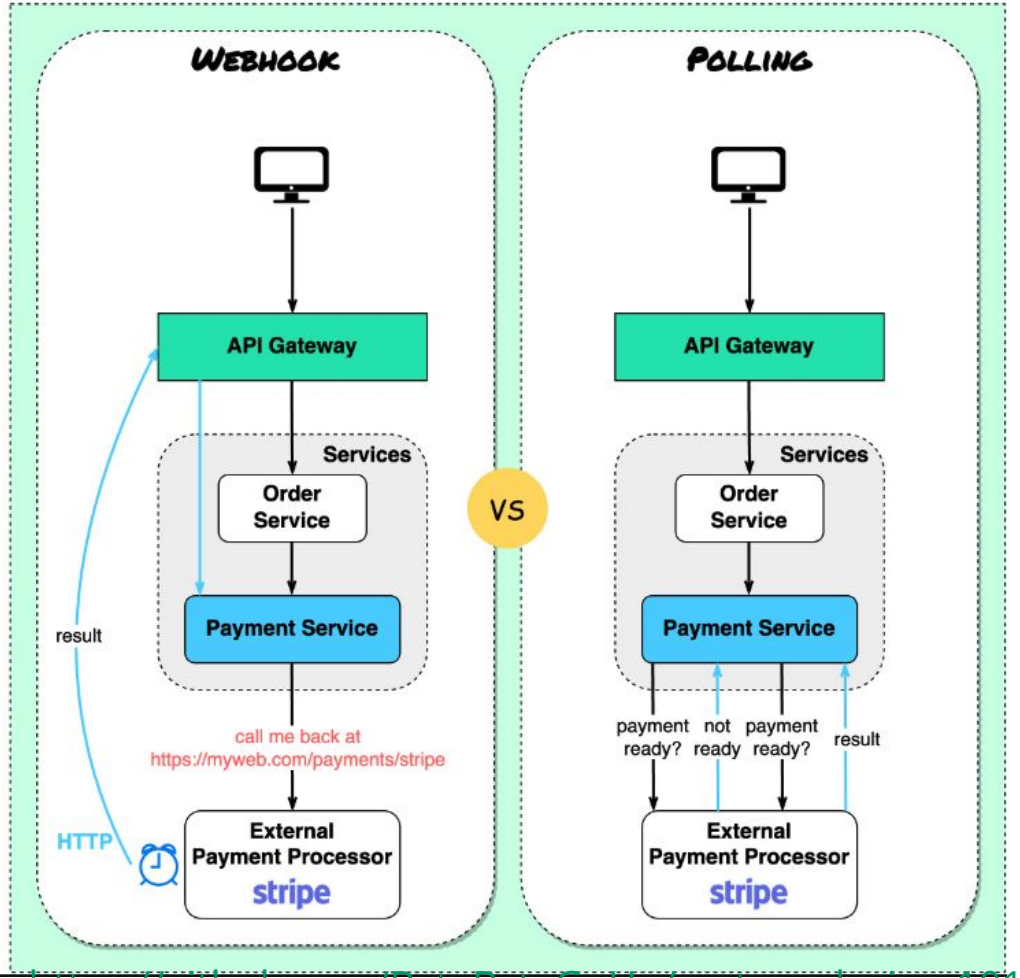
  Webhook可以用於將數據傳送到分析工具中。例如將 Google Analytics中的事件數據傳送到第三方分析工具，進行更深入的分析。



例子

- 客戶端透過API Gateway向訂單服務發送訂單
- 訂單服務再到支付服務進行支付交易。
- 支付服務與外部支付服務提供者(PSP) 溝通以完成交易。

https://github.com/ByteByteGoHq/system-design-101/blob/main/images/webhook.jpeg



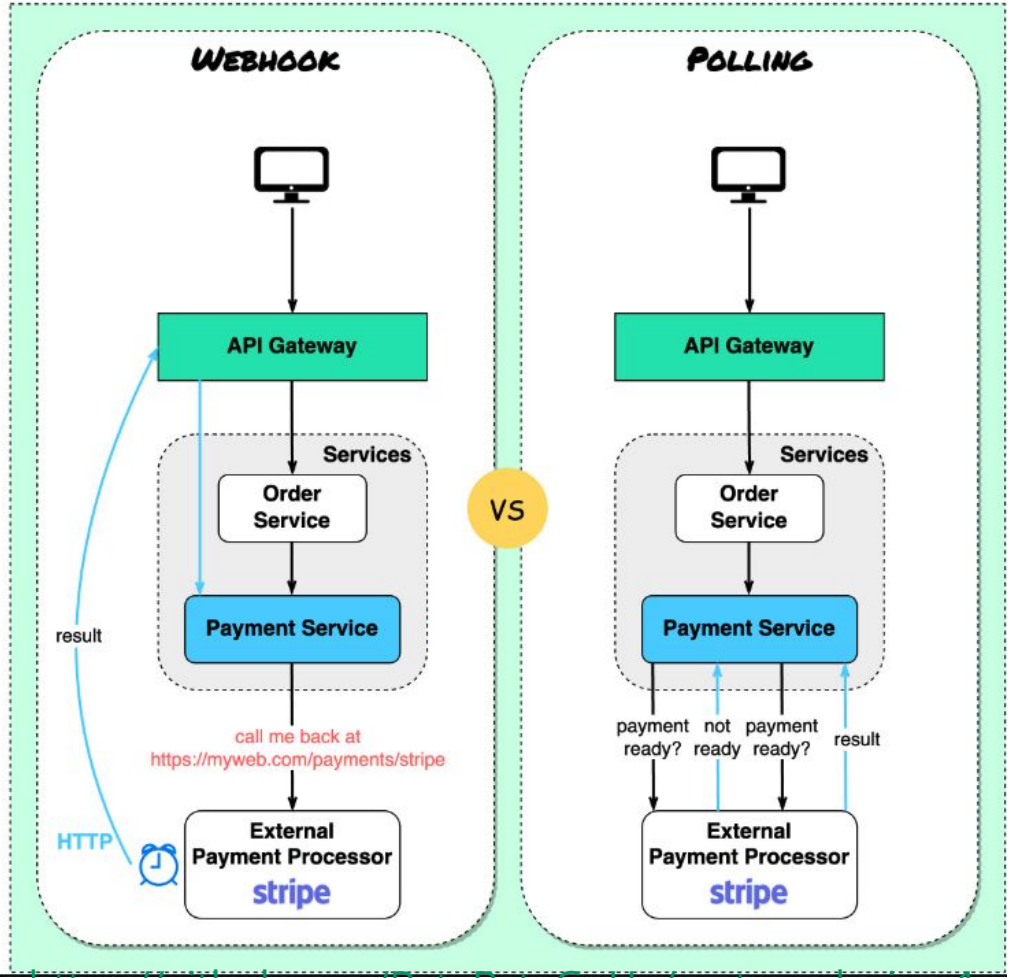
- Polling

支付服務向payment service provider (PSP)發送付款請求後，不斷向PSP詢問付款狀態。
經過幾輪之後，PSP終於回覆狀態。

缺點：

1. 持續輪詢狀態需要支付服務的資源。
2. 外部服務直接與支付服務溝通，容易產生安全漏洞。

https://github.com/ByteByteGoHq/system-design-101/blob/main/images/webhook.jpeg



https://github.com/ByteByteGoHq/system-design-101/blob/main/images/webhook.jpeg

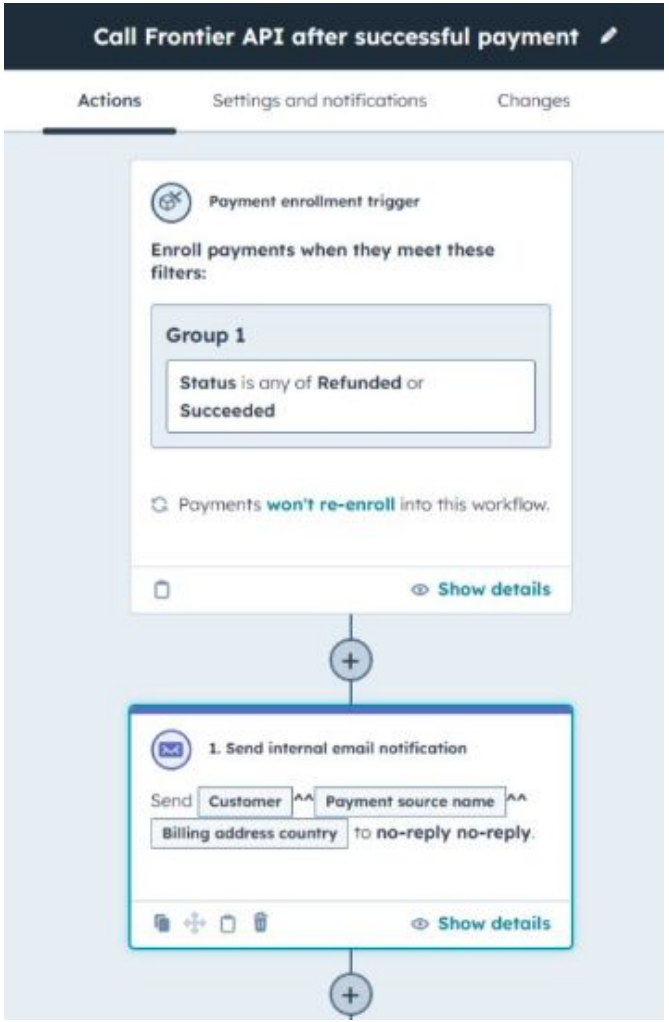
- Webhook（伺服器向客戶端發送 HTTP 請求）

我們可以向外部服務註冊webhook。當有請求更新時，call back某個 URL 。

當 PSP 完成處理後，invoke HTTP request來更新付款狀態。這樣支付服務不再需要浪費資源來輪詢支付狀態。

- 可以設定排程（每小時檢查是否更新付款狀態。）

1. 設計一個合適API供外部服務呼叫。
2. 需要在 API Gatway中設定適當的規則。
3. 需要在外部服務中註冊正確的URL。



官網和Sass平台溝通，實作資訊交換

Eg.

1. 使用Webhook，當客戶在官網付款成功後 Trigger API

2. 將付款人的名稱和購買方案寄到公司 email 信箱

3. 在Sass平台註冊幫付款成功的使用者註冊帳號密碼

4. 公司email 信箱寄信給使用者帳號密碼

當客戶在官網付款成功後 => Trigger Frontier API幫客戶在平台建立帳號密碼

Frontier

摘要

Designer Plan - Designers & Brands 每月 $39.00

總計 $39.00

後續 每月 $39.00

連絡資訊

電子郵件地址 *

付款資料

金融卡 美國銀行帳戶

卡號

1234 1234 1234 1234

有效期 CVC

CVC

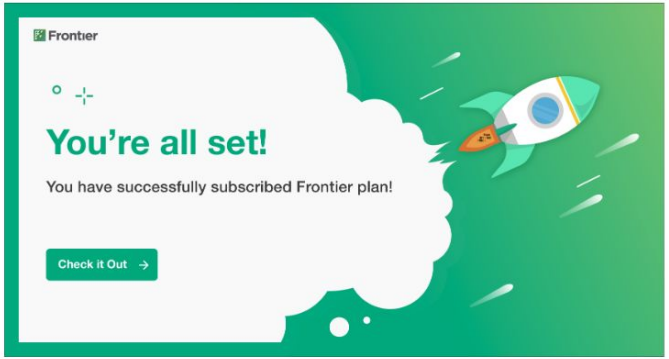
Frontier

你來自 Frontier.cool Inc. 的收據

付款	付款日期	付款方式
$39.00	2024年3月7日	VISA - 5514

產品	數量	總計
Designer Plan - Designers & Brands 2024年3月7日 - 2024年4月7日 的收費	1	每月 $39.00

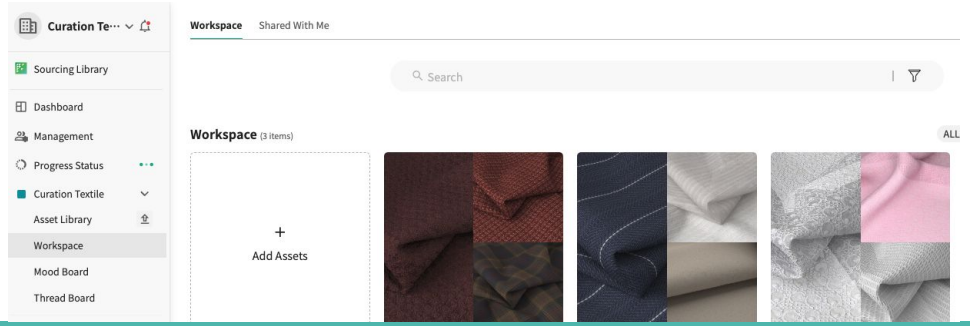
Frontier



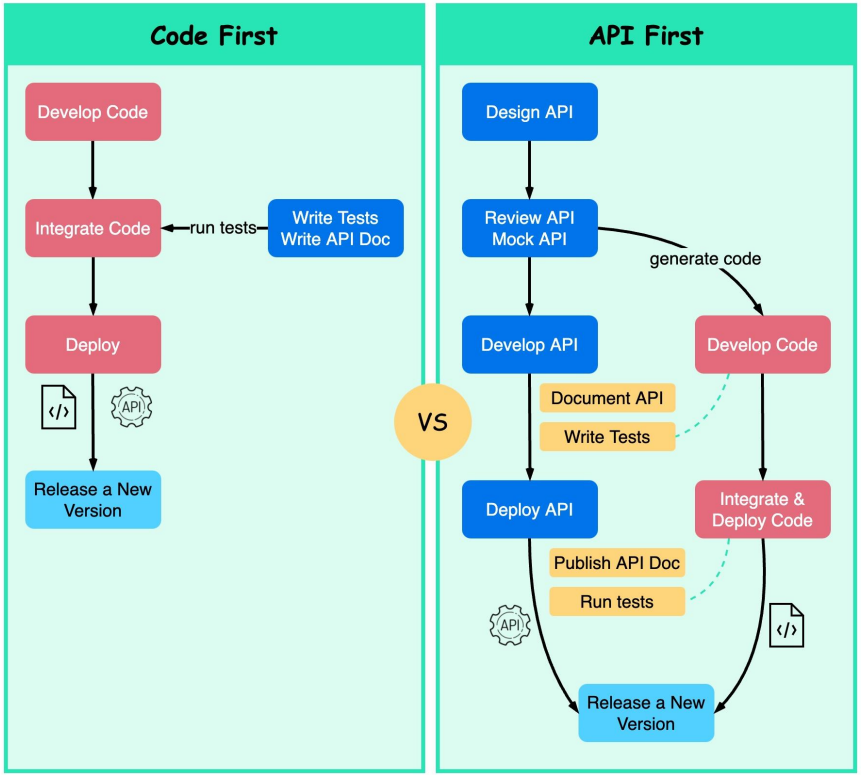
Great news!
You have successfully subscribed your designer plan!

Hi there!

We're so excited to let you know that you have successfully subscribed designer plan for **tingtinghsu.tw_2509**



Code-first vs API-first



- code-first development

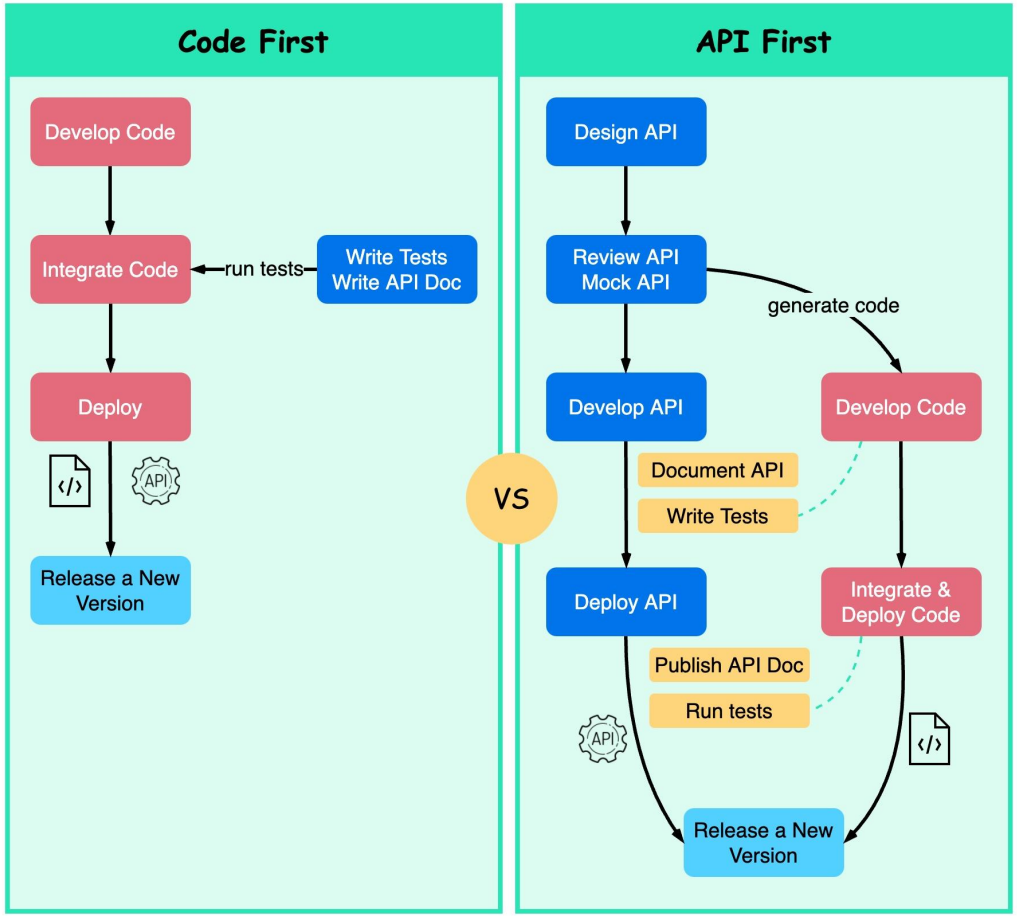
（著重於應用程式的邏輯和功能）

先撰寫應用程式的代碼，然後再設計和開發相關的API。

開發人員通常會先考慮應用程式的邏輯和功能，然後再根據這些需求來設計API。

好處：開發人員可以更容易地控制和調整應用程式的邏輯，並且可以根據需求進行靈活的開發。

https://github.com/ByteByteGoHq/system-design-101/blob/main/images/api_first.jpg

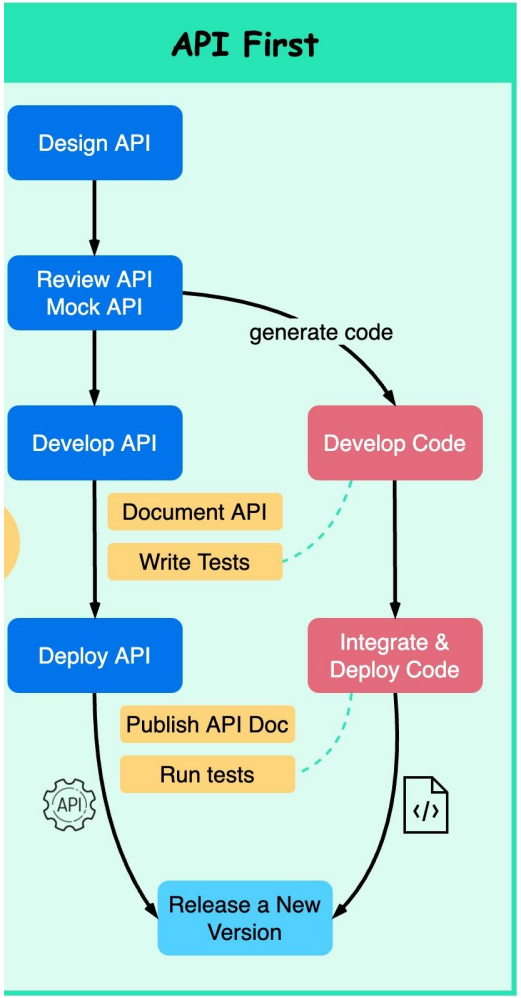


WHY API-first?

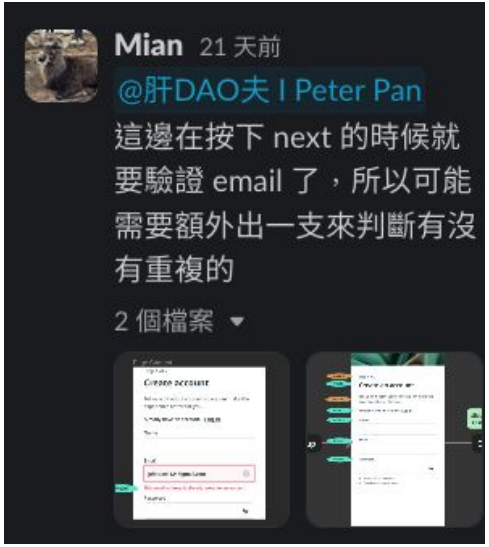
- API-first development（則著重於API的設計和功能）是一種以API為中心的開發方法。它的重點是先設計和開發API，然後再根據API的規範來撰寫應用程式的代碼。

- 強調了API的重要性，使得開發人員可以更好地設計和定義API的功能、輸入輸出和資料格式等。

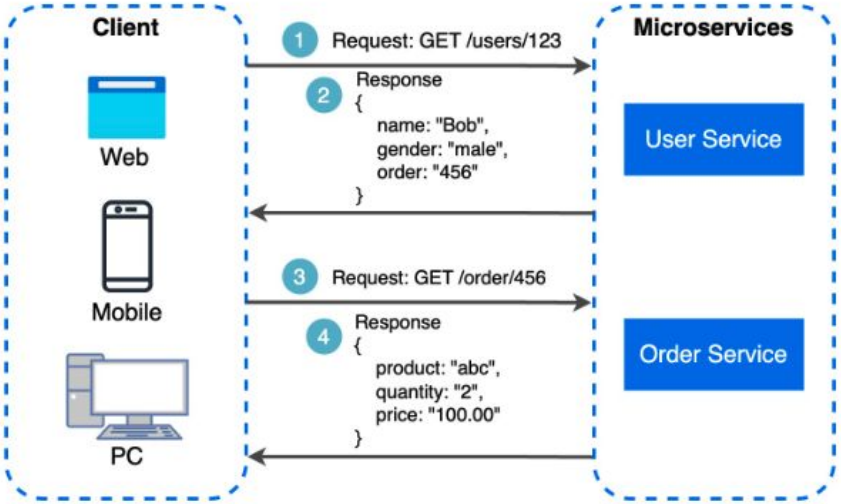
- 有助於提高應用程式的可擴展性和互操作性。



Aa Name	route
一般註冊	/sign-up/general
Google 註冊	/sign-up/google
檢查該email是否已註冊	/sign-up/check-email-exist
一般登入	/sign-in/general
Google 登入	/sign-in/google
取得 User 資料	/user
記錄使用者透過何處得知 FabriSelect	/user/how-to-discover
切換語系	/user/change-language
忘記密碼寄信	/user/forgot-password/send-email
忘記密碼驗證	/user/forgot-password/verify
忘記密碼修改	/user/forgot-password/reset-password
取得 Org 資料	/org/get
取得使用者在組織下的資料	/org/user/get
取得 Group 資料	/org/group/get
取得使用者在團隊下的資料	/org/group/user/get



REST API vs. GraphQL



REST API（Representational State Transfer API，表現層狀態轉換API）

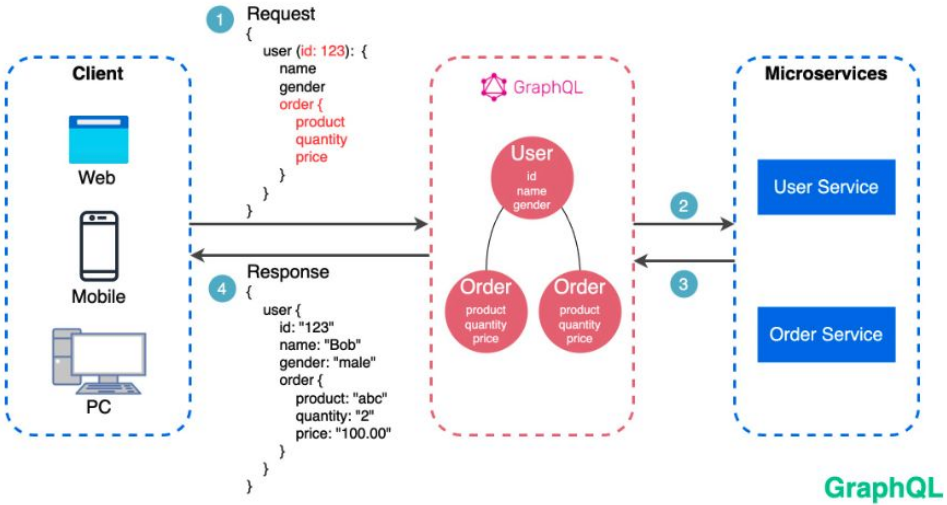
使用HTTP協議來進行通信，通過URL定位資源，使用HTTP方法（如GET、POST、PUT、DELETE）來操作資源，使用不同的狀態碼和資源表述來表示操作結果。

流程：

- 用戶端向伺服器傳送請求。用 戶端按照 API 文件，以伺服器能理解的方式格式化請求。
- 伺服器對用 戶端進行身分驗證，並確認用 戶端有權發出該請求。
- 伺服器接收請求並在 內部處理。
- 伺服器向用 戶端傳送回應。回應包含告知用 戶端請求是否成功的資訊。該回應還包含用 戶端請求的任何資訊。

https://github.com/ByteByteGoHq/system-design-101/raw/main/images/graphQL.jpg

WHY need GraphQL?



GraphQL *a server-side runtime and query language for your API*

由Facebook開發的查詢語言和執行引擎，允許客戶端精確地指定需要的資料，並返回符合指定結構的結果。

與REST API相比，GraphQL具有以下優勢：

- 提供更高的靈活性，客戶端可以根據其需要精確地指定所需的資料，避免了過度載入或多次請求的問題
- GraphQL具有強大的型別系統，使得開發人員可以準確地定義資料模型和查詢結構。
- GraphQL支援批量操作和實時數據更新，使得應對複雜的資料需求和即時應用變得更加容易。

RESTful 最大的好處之一就是遵循 HTTP 規範，因此只要單一 end point 有問題，就可以透過 statue code (5xx) 知道問題發生。

而 GraphQL 如果在 client 端發生問題的話，也只會回 200，所以對於錯誤捕捉可能需要花費額外的功夫，例如去監聽 client side 回傳回來的 message，但這個也需要 client 去額外處理。

https://github.com/ByteByteGoHq/system-design-101/raw/main/images/graphQL.jpg

Thank You!