

OPEN API

潘彥勛

pan820124@gmail.com

大綱

- ▶ RESTful API
- ▶ Open API
- ▶ Demo

什麼是 RESTful API?

RESTful API 是什麼？

► 就是我們很常用的那個阿

- REST (Representation State Transfer) API 本質上是一種「設計風格」
- 有以下幾種特徵：



HTTP 架構



資源導向



無狀態



統一接口

到底是 REST API 還是 RESTful API?

► 哪個是正確用法

遵循 REST 架構風格的 API
稱為 REST API

實作 REST 架構的 Web 服務
稱為 RESTful Web API



但現在大家都混用了
REST API = RESTful API

RESTful API 的設計要點

► 就像在講一句話

- URL 的設計要針對資源設計
- HTTP Method 告訴服務端你想做什麼
- 由 HTTP 標準狀態碼宣告結果
- 如果有，會回傳固定格式的資料

/users/{id}

針對一個 user ID

GET

取得資料

200

執行狀態 OK

{JSON}

取得的內容

HTTP Method – 用來描述要做什麼動作

HTTP Method	Description
GET	讀取資源
POST	建立資源
PUT	更新全部資料
PATCH	更新部分資料
DELETE	刪除資料

常見問題 – 以下三種哪個是 RESTful 風格

► 情境：針對一個 User 取得資料

`/getUsers/{id}`

GET

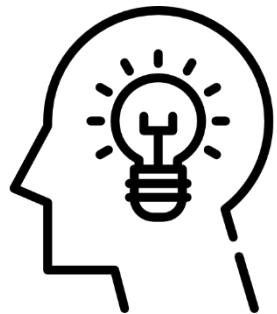
`/getUsers/{id}`

Post

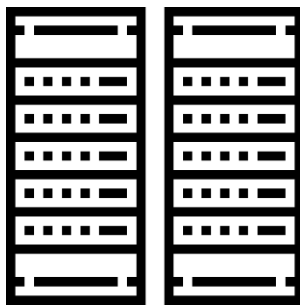
`/users/{id}`

Post

使用 RESTful 設計有什麼好處？



易於理解



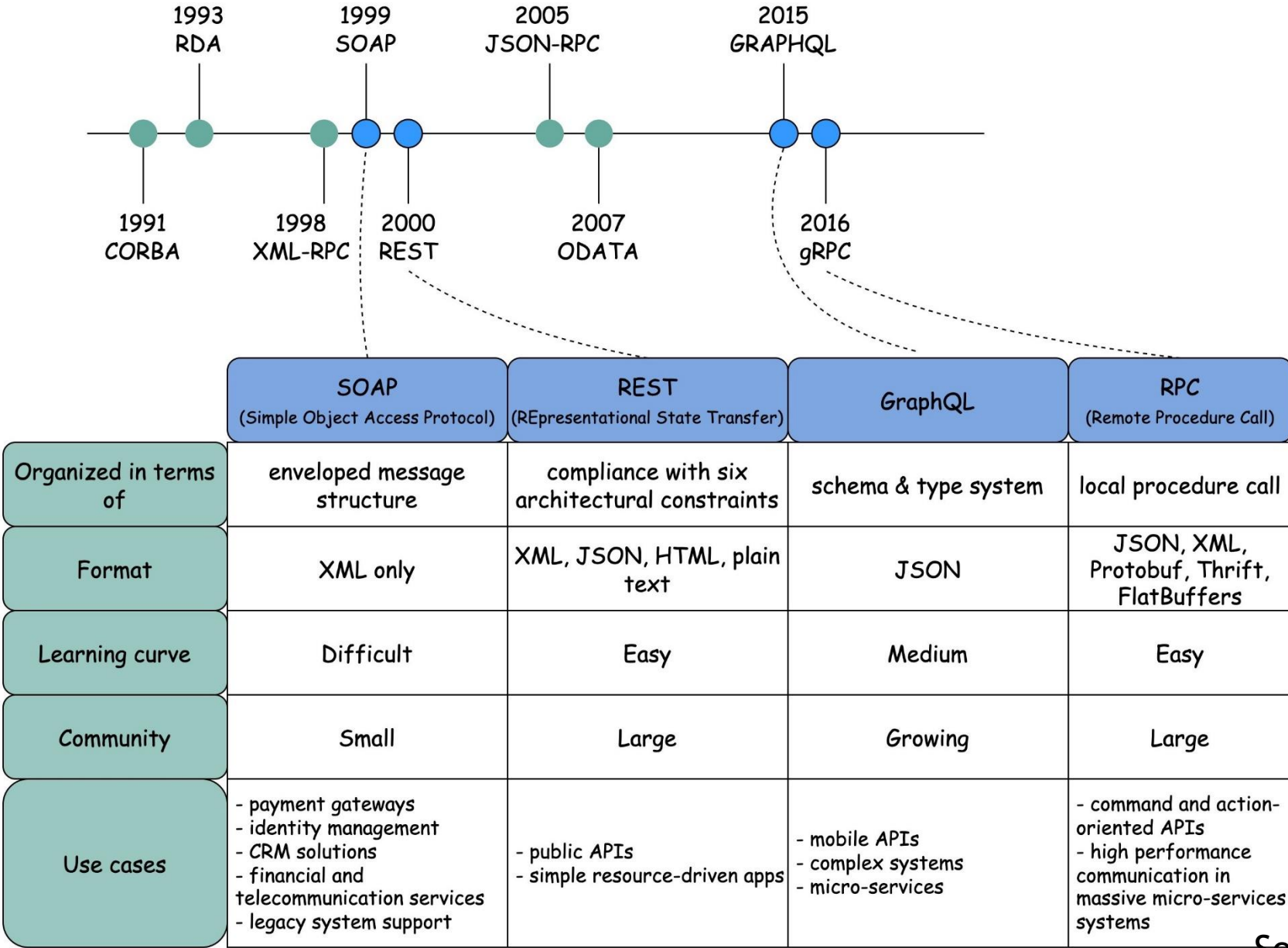
可擴展



跨平台

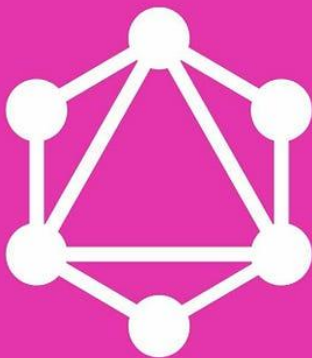
主流 API 比較

主流 API 發展





REST



GRAPHQL

 **RPC**

GRPC

RESTful API, gRPC, GraphQL 差異比較

特性/用途	RESTful API	gRPC	GraphQL
傳輸協議	HTTP/1.1	HTTP/2	HTTP
資料格式	通常為 JSON 格式	Protocol Buffers	通常為 JSON 格式
設計導向	將每個 URL 映射為資源 使用標準 HTTP Method (GET, POST, PUT, DELETE)	方法導向，使用 gRPC 服務和方法的遠端調用	查詢導向， 使用一個端點來定義查詢
強類型支持	無內建	內建	手動定義
	性能適中，主要依賴於 JSON 的解析	高性能，特別適合高效能系統，低延遲	對小型查詢性能高效，查詢可針對具體需求精簡
狀態管理	無狀態	支持流式傳輸和雙向流式通信	無狀態
上手門檻	低	高	中
適用情境	適用於大部分網路應用或簡單 HTTP 請求的系統	適合內部系統通信或微服務	適用於需要靈活查詢、精確數據提取的應用系統

Top 8 API Architectural Styles

SOAP



XML-based, strict, favored in enterprises

REST



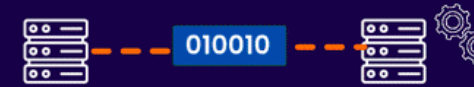
Building Scalable APIs using standard HTTP Methods.

GraphQL



Query Language for APIs, Allowing Clients to Request Specific data.

gRPC



High Performance Framework For Remote Procedure Calls, Using HTTP/2.

WebSocket



Bidirectional, Real Time Communication Protocol.

Webhook



Event-Driven, Server-Side Mechanism that sends HTTP Callbacks.

MQTT



Lightweight messaging protocol for constrained devices.

AMQP



Open messaging protocol for reliable communication.

Restful API 缺點是什麼

- RESTful 是個設計風格，並不是強制的
- 如果有不合規設計，也是能動，所以會有誤用
- 想對這些設計進行規範，於是出現了 Open API

Open API & Swagger

什麼是 Open API

OpenAPI 是一種規範，用於定義和描述基於 HTTP 的 API，主要用於規範與加強 RESTful API。

在 OpenAPI 中，「定義 API 結構」是指通過 Open API 規範來詳細描述 API 的各個組成部分，包括：

- URL
- HTTP Method
- Request Param
- Response
- Security

這些描述使用 JSON 或 YAML 格式編寫，使 API 的結構清晰明了，也方便生成文件、自動化測試和生成對應程式碼

跟 Swagger 有什麼關係

最初，Swagger 是一個包含了 API 規範和工具的開源項目，用於設計和規範 RESTful API

隨著時間推進，Swagger 規範獨立出來改名為 OpenAPI 規範，並由 OpenAPI Initiative 管理。

現在所說的 Swagger 通常指的是工具集



工具集



規範

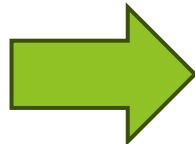
Open API 規範 – 結構定義

Open API 規定了一份文件的基礎結構：

- Info: 關於 API 文件資訊
- Servers: 提供 API 服務的伺服器連結列表
- Path: 描述 API 的各個端點及其 HTTP 方法，也是文件的核心
- Components: 對於會重複利用的資料結構，可以定義成 Model

文檔的好處是什麼？

- ▶ 一份文檔可以有多種功效



Swagger UI

Supported by SMARTBEAR



Swagger Codegen™

Supported by SMARTBEAR



Swagger Editor™

Supported by SMARTBEAR



POSTMAN

Swagger 工具們

► 就這三個



Swagger UI

Supported by SMARTBEAR



Swagger Codegen™

Supported by SMARTBEAR



Swagger Editor™

Supported by SMARTBEAR

Swagger UI - 閱讀、預覽、直接用

► 不用寫程式就可以用了

Swagger UI 直接將 yaml 轉成 UI 介面

讓使用者可以更直觀的填寫參數與呼叫

對於團隊中非開發人員而言，可以更直接的使用

同時也像是一份線上說明文件



Swagger Editor

► 提示你規範怎麼寫

Swagger Editor 可以提示使用者是否有依循 Open API 規範開發

像是驗證語法、即時使用 UI 等

對於開發者來說很方便的工具



Swagger Codegen

► 使用 Java 開發的程式碼生成工具

Swagger Codegen 能為使用者/開發者生成基本的「程式骨架」

對於開發團隊需要支援多種語言的時候

一個 yaml 就可以快速產出多種語言的框架

對開發人員來說將功能專注在邏輯的處理



Swagger Codegen 支援列表

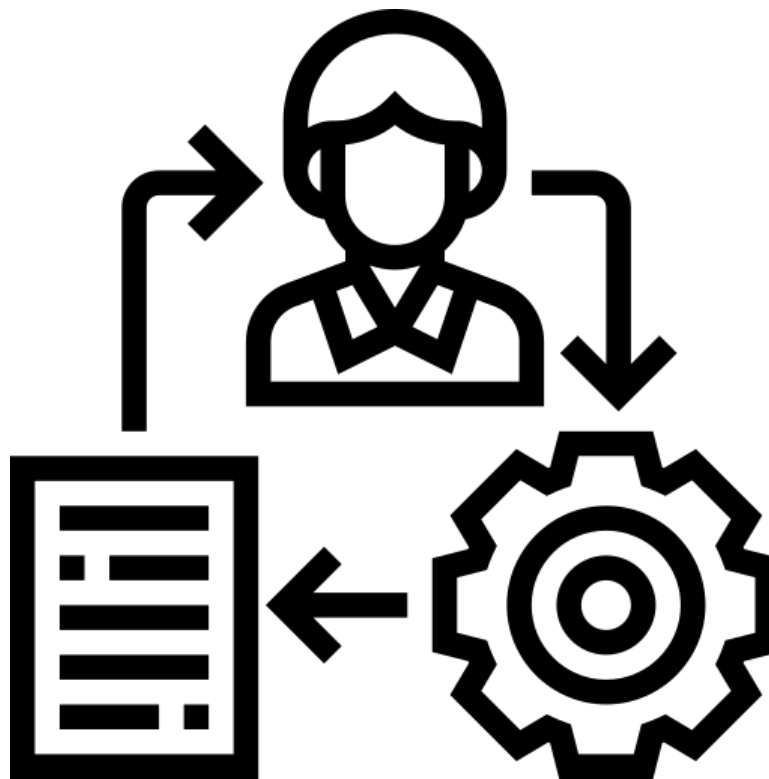
- ▶ 支援但不限以下語言與框架

Server Side	Client Side
Java (Spring Boot)	Java
Python (Flask 、 Django)	Python
Node.js (Express)	TypeScript (Angular 、 Axios)
Go	Go
.NET Core	C#

回顧

- 一份文檔可以有多種功效

For PM:
可讀的規範化文檔
也可轉成可用的操作介面



For PG:
自動生成程式碼可以加速開發
可以版控

For QA:
可以透過自動化工具進行測試

QA

Thank you!