

backend 課程（ 第一課 ）

by

Triton Ho



今天大綱

- 淺談 MVC
- 淺談 backend 基本概念
- 介紹 golang
- 從菜鳥亂來的 CRUD 到 backend framework
- 介紹 Application Server 中主要模組

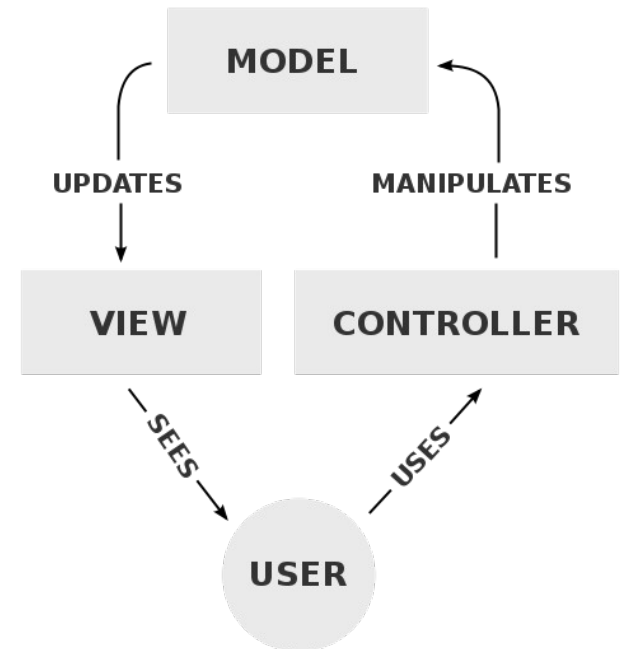
淺談 MVC

淺談 MVC

- 最初是 SmallTalk 的概念，在 8 0 年代開始流行
- 針對 Standalone software，而不是今天的 Server-Client Model

MVC 中的角色

- Model
 - 系統中各式各樣的抽象物件
 - 例如：User、Cat、CatFood
 - Model 改動時，會主動告知 View 需要重新顯示
- View
 - 把 Model 顯示給用戶看
 - 有控制項，能讓用戶輸入資訊和行動並且傳給 Controller
- Controller
 - 根據用戶輸入和行動，和商業邏輯對 Model 作出更改



By RegisFrey - Own work, Public Domain,
<https://commons.wikimedia.org/w/index.php?curid=10298177>

M+V+C != MVC

- 把抽象物件 (Model Class, M)、顯示層 (Presentation tier, V) 和控制程式碼 (Business logic tier, C) 分開來寫，只代表你的系統中有 M+V+C
- M+V+C 不等於 MVC 的
 - 你硬說你的 M+V+C 也是 MVC，改變了 MVC 的本來定義也不是不行。一切都只是名字而已，反正太陽餅也沒有太陽成份
 - 所以，每家 framework 對 MVC 的定義都是不同的，因為大家都自行把 MVC 再定義了

MVC 跟 HTTP 的不相容

- 在 MVC 中，Model 改動後需要以 Observer Pattern 去主動告知 View 去作出對應更新；但是 HTTP 世界中，我們極少進行 Server Push；而是由 Controller 回應給 View
- MVC 針對的是單機軟體，所以 M, V, C 之間以 function call 的方式溝通沒有任何問題；而 HTTP 是針對網路，一般都使用 RESTful 方式通訊

對 MVC 結語

- 單純地說自己正在使用什麼 framework 便好了，別再提 MVC 這三個字了
- 近年越來越流行 frontend javascript framework，伺服器作好 API endpoint 後返回 json / xml 數據便好
 - Presentation layout 越來越跟 backend 沒關係了

淺談 backend 基本概念

- backend 最看重的是 throughput (系統總容量) , 每一 Request 的處理速度相對上不太重要
 - 當你的 execution time 壓低到某程度後 , network latency 會變成主要因素
- $\text{Throughput} = \text{concurrent execution} / \text{execution time per task}$
- HTTP Request 基本上都是 Embarrassingly parallel problem
 - 例子 : 我在 facebook 發文 , 跟你在 facebook 發文 , 其所涉及的 Data 是完全沒相關的
- 對 Application Server 來說 : 增加機台便能無痛增加 throughput
- 而 RDBMS 只有單一 Master , 通常會成為系統效能瓶頸

Strongly typed vs Weakly typed

- 對於很小心的老手，WT 寫起來會快一點點
- ST 能讓你在 compile 時便立即看到自己的低級腦殘錯誤
- ST 會多花了時間在 Coding，但是少花了時間在 Testing 上
- 對有一定操守的開發者，其實真正會吐血的一般都是 Race Condition / Design Flaw
 - 不管是 ST 和 WT，對於這些問題都沒有太大幫助
 - 所以，挑你喜歡的語言吧，ST / WT 不是重點

對 OOP 雜感

- 多年撲火感想：OOP 跟 Good code 是.....
 - not correlated
 - 我看過了一個 10000 行的 class；也看過優美而精準的 PL/SQL
 - Good code 看開發者的功力和修養，還有看老闆給的開發時間
- Getter 和 Setter 不一定是好事耶
- 主流的 programming language，幾乎沒有純粹的 OOP
 - C++ / Java 的 int 不是物件耶
- 即使不支持 OOP，今天大部份語言都支援 object-based programming
 - C 有 struct

介紹 golang

- 語法上，相對接近 Scripting language
- 極端性防呆
- Strongly typed language
- 不支持 OOP，但是支持 Interface
- 支持 struct tag
- 內建 channel
- 內建 package 管理功能
- 內建基本的測試功能

Application Server 中主要模組

- Router
- Middleware
- Auth
- Input Binder
- Basic Validator
- Transaction Manager
- Database connector / ORM
- Output Utility
- MQ connector
- Logger
- File Manager

完