

Rails Migration & Model 欄位設計原則

Kait Wang @ WTT 2016.09.08

About Me

- Kait Wang
- Blog: <https://sibevin.github.io/>
- uknowiknow.com 你知我知好學網

#1 時間欄位使用「動詞過去式_at」來命名

- 只要看到「動作過去式_at」的欄位就知到是時間欄位
- 為什麼不用進行式或名詞？

#2 在 migration 中使用 null: false

- 除非欄位一直都會必填才加
- 在 model 中加上 `presence` validation
- migration option 與 model validation 的差別？

#2 在 migration 中使用 null: false

- 避免因為資料不存在而造成的bug
- 減少不必要的「檢查欄位是否有值」的動作，因為一定值

#3 在 migration 中使用 default

- 除非明確知道欄位的預設值，而且不會變動才加
- 會變動的預設值可以使用 `default_value_for gem` 在 model 中做設定

#4 在 migration 中使用 limit

- 除非明確知道欄位的長度，而且不會變動才加
- 使用 model 的 `length / format validation`
- integer 的 limit

#5 在 migration 中使用 unique index

- 如果欄位的值是唯一，則可以加 unique index
- 避免使用 model 的 unique validation
- 因為不加 unique validation，所以要記得做 exception handling

#5 在 migration 中使用 unique index

unique validation 的問題

- race condition
- 寫入資料的效能問題

#6 關聯用的欄位名稱一定是 「_id」結尾，而且要加 foreign_key: true

- rails generator 可以使用 references 關鍵字
- 視情況加 null: false 與 index: true

#6 關聯用的欄位名稱一定是 「_id」結尾，而且要加 foreign_key: true

Convention over Configuration (慣例優於設定)

在 _id 前的名稱不一定是 model 的名稱，可以用
association 的 class_name 與 foreign_key 來調整

#7 使用 enumerate

- 當欄位只用來記錄某個值，而且這個值已經事先定義好，通常用於下拉選單或是 radio button
- 建議使用 integer 並設成 limit: 1

#7 使用 enumerate

使用 `enumerate` 的好處

- integer 欄位在 query 上比較有效率，同時節省欄位空間
- 在程式碼中仍然是用字串來設定欄位，增加程式的可讀性
- 本身支援 `simple_form`，只要一行程式就幫你生 `select` 與 `radio button`
- 支援 `i18n`

#8 其它好用的gem

- `uidable` - 欄位的值需要在 model 被建立時產生，而且之後這個欄位只能讀取而不能修改
- `aasm` - 處理狀態機

#8 其它好用的gem

- `acts_as_list` - 記錄資料的排序
- `awesome_nested_set` - 處理樹狀結構

Q & A