

題目: https://universetech.hackmd.io/@golden/full_stack
(Backend)

答案請從第 4 頁開始

1. 請實作下列需求

假設目前有個需求，需要透過多個號源站抓取各彩種指定期號的中獎號碼
可使用 **Laravel** 或您熟悉的程式語言，請設計出架構，實現剛好滿足需求並
具有擴充性程式碼

- 規格如下
 - 需主號源及任一副號源比對相同才成功（主號源只有一個，副號源將會有多個）
 - 不同彩種會有不同主號源需求
 - 重慶時時彩，請以號源一為主號源
 - 北京 11 選 5，請以號源二為主號源
 - 有擴充彩種及號源站需求

附註：

彩種為彩票的供應商，例如 重慶時時彩、北京 11 選 5 等等
號源為第三方 API 廠商，整合各彩種開獎號碼

- 使用方式

```
<?php
class UpdateWinningNumberJob
{
    protected $lottery;

    public __construct(Lottery $lottery)
    {
        $this->lottery = $lottery;
    }

    public function handle()
    {
        try {
```

```

        // $this->lottery->gameId 為 int，指彩種編號，
        // 重慶時時彩 = 1
        // 北京 11 選 5 = 2
        // $this->lottery->issue 為 string，為此 lottery 期號 (e.g.
        "20190903001")

        $target = new xxxx($this->lottery); // 請實現此 class
        $this->lottery->update([
            'winning_number' => $target->getWinningNumber();
        ]);
    } catch (FetchFailureException $e) {
        Log::error('Something went wrong.');
```

- 號源一 API (為模擬簡化，非正式 API)
 GET <http://one.fake/v1?gamekey={gamekey}&issue={issue}>
 gamekey 為彩種編號
 issue 為開獎期號

gamekey 對照表

重慶時時彩 => ssc

北京 11 選 5 => bjsyxw

```

// GET http://one.fake/v1?gamekey=ssc&issue=20190903003
{
    "result": {
        "cache": 0,
        "data": [
            {
                "gid": "20190903003",
                "award": "0,6,2,2,3",
                "updatetime": "1567446006"
            }
        ]
    },
    "errorCode": 0
}
```

- 號源二 API（為模擬簡化，非正式 API）
GET <https://two.fake/newly.do?code={code}>
code 為彩種編號

code 對照表

重慶時時彩 => cqssc

北京 11 選 5 => bj11x5

```
// GET https://two.fake/newly.do?code=cqssc
{
  "rows": 3,
  "code": "cqssc",
  "data": [
    {
      "expect": "20190902003",
      "opencode": "3,8,1,9,5",
      "opentime": "2019-09-02 01:12:46"
    },
    {
      "expect": "20190902002",
      "opencode": "3,1,5,8,6",
      "opentime": "2019-09-02 00:52:37"
    },
    {
      "expect": "20190902001",
      "opencode": "6,1,9,0,3",
      "opentime": "2019-09-02 00:32:03"
    }
  ]
}
```

假設今天要 query 的是重慶時時彩，且期號為 201909030012，則 url 如下：

<http://localhost/Lottery/index.php/Main/start?gameId=1&issue=201909030012>

```
← → ↻ ⓘ localhost/Lottery/index.php/Main/start?gameId=1&issue=201909030012
🔗 應用程式 🔗 URL to PDF Conve...

gameId=1 => 重慶時時彩
gameId=2 => 北京11選5

-----
主號源API url: http://localhost/Lottery/index.php/API1/query?gamekey={gamekey}&issue={issue}
Submit http get request to 「http://localhost/Lottery/index.php/API1/query?gamekey=ssc&issue=201909030012」 for retrieve data...
response={ "result": { "cache": 0, "data": [ { "gid": "201909030012", "award": "0,6,2,2,3", "updatetime": "1567446006" } ] }, "errorCode": 0 }
開獎號碼 = 0,6,2,2,3

-----
從其他副號源搜尋是否有任一個的開獎號碼跟主號源一樣!
Submit http get request to 「http://localhost/Lottery/index.php/API2/query?code=cqssc&expect=201909030012」 for retrieve data...
response={ "rows": 1, "code": "cqssc", "data": [ { "expect": "201909030012", "opencode": "0,6,2,2,3", "opentime": "2019-09-02 01:12:46" } ] }
開獎號碼 = 0,6,2,2,3

-----
開獎號碼: 0,6,2,2,3
```

假設今天要 query 的是北京 11 選 5，且期號為 201909180023 則 url 如下：

<http://localhost/Lottery/index.php/Main/start?gameId=2&issue=201909180023>

```
← → ↻ ⓘ localhost/Lottery/index.php/Main/start?gameId=2&issue=201909180023
🔗 應用程式 🔗 URL to PDF Conve...

gameId=1 => 重慶時時彩
gameId=2 => 北京11選5

-----
主號源API url: http://localhost/Lottery/index.php/API2/query?code={gamekey}&expect={issue}
Submit http get request to 「http://localhost/Lottery/index.php/API2/query?code=bj11x5&expect=201909180023」 for retrieve data...
response={ "rows": 1, "code": "bj11x5", "data": [ { "expect": "201909180023", "opencode": "0,6,2,2,3", "opentime": "2019-09-02 01:12:46" } ] }
開獎號碼 = 0,6,2,2,3

-----
從其他副號源搜尋是否有任一個的開獎號碼跟主號源一樣!
Submit http get request to 「http://localhost/Lottery/index.php/API1/query?gamekey=bjsyxw&issue=201909180023」 for retrieve data...
response={ "result": { "cache": 0, "data": [ { "gid": "201909180023", "award": "0,6,2,2,3", "updatetime": "1567446006" } ] }, "errorCode": 0 }
開獎號碼 = 0,6,2,2,3

-----
開獎號碼: 0,6,2,2,3
```

Source code 請參考 github，程式進入點 controller source code 在這個路徑：

Lottery\application\controllers\Main.php

本系統為了模擬情境，自己寫了 API1.php, API2.php，假設是真的 query 完要回傳的 json response format。

2. 上面程式或規格可能存在什麼潛在問題？還可以怎樣優化？

1. 假如程式沒寫好的話，可能會有一堆 if else 的判斷式，每當要擴充號源的時候，程式就必須修改。
2. 由於每一家 api response 的 json format 都不一樣，甚至哪一天有可能被改成回傳的是 xml format，又或者新加入的擴充 api 回傳的又是其他 format，因此本程式將各種 api 的 parse rule 拉成 global variable，每當有新的號源要加入，則 maintain 該對應的 parse rule 即可，當然如果可以把 global variable 改成從 ap config(server 上的 config file or DB 的 config table)讀取，就可以達到完全不修改程式碼的情況下進行擴充！

3. 如果要加入第三家號源，會怎麼進行擴充？

程式已設計成可透過修改 global variable 就進行擴充，如程式中的 \$gameld_mainSource_mapping 加入新的彩種 id 以及 api url with params.

程式中的 \$source_gameld_mapping 加入新的 api url 以及彩種對應的彩種編號

程式中的 \$source_parse_result_rule_mapping 加入新的 api url response data 的 parse rule.

當然如果可以設計成用 ap config 的方式就更好了，如在 server 上的一個 ap config file 或者是從 DB 的某一張 config table，程式透過讀取設定方式，即可達成擴充，且不必修改程式碼！

4. 每個號源有不同的速率限制，會如何實現限流，防止被 ban？

假設號源一限制 5 秒一次，號源二限制 3 秒一次

同號源不同彩種是分開計算的

程式中要有判斷機制，需要紀錄上次 request api url 的時間和彩種，然後判斷這次要 query 的彩種在對應的 api url 拿出上次的 request 時間來判斷，看看是否有超過該 api 的速率限制，有的話就禁止發送這次的 request!

例如 user 這次要 query 彩種編號 1(重慶時時彩)，由於彩種編號 1 的主號源是 api 1，所以會先 request 這個 api 去拿資料，程式判斷到這個 request url，彩種編號 1 上次 query 時間是 2019/10/04 09:00:00，目前時間是 2019/10/04 09:00:04，已經超過在 5 秒內的限制，為了防止被 ban，因此就不送出此次 request 了!

5. 開獎時間並非準時，您會如何實現重試機制？

號源站並非能第一時間抓到該彩種中獎號碼，因此存在抓不到的可能性

可以起個 cronjob 去每 n 秒 trigger 一次自己寫的這隻程式，一旦該 api 已經有中獎號碼的資料的時候，就可以馬上拿到中獎號碼了!

由於上題已經提到可以從程式設計機制來避免被 ban，因此也不用擔心如此會有問題!

6. 可以實現哪些手段來減少程式運行時間？

可以起 multiple thread 同時去 request 各個 api，當比對到主號源跟任一副號源中獎號碼一樣時，就回傳結果!