I. Một số hàm dùng trong driver MongoDB của PHP:
   1. Tìm kiếm:
   2. Update:

II. Mẹo vặt:
   1. Xem các method của class: Lumen có sử dụng các thư viện của Laravel. Để tận dụng tối đa quy tắc DRY (Don't repeat yourself), Laravel sử dụng khá nhiều Magic Method của PHP, do vậy đôi lúc chúng ta gọi 1 class mà IDE (như PHPStorm, Netbeans,..) không thể gợi ý cho chúng ta biết class đó có những method nào. Để xem đầy đủ method của 1 class, ta sử dụng hàm PHP: get_class_method($class) trong đó $class có thể là tên class hoặc là 1 object.
   2. Debug với print_r($params);die;
   3. Adapter truyền dữ liệu qua json, do vậy để dễ xử lí, ta thường dùng hàm json_decode($json) để chuyển chuỗi json thành array.
      Trước khi lấy bất cứ field nào của json (dưới dạng array), ta nên dùng hàm isset() để kiểm tra key của array có tồn tại hay không.
      Ví dụ:

```
$arrJson = json_decode($json, true);
if ($arrJson['OrderNumber']) {
 // do something
}
```

      Code như trên quá nguy hiểm vì rất có thể json lấy về không hề có field OrderNumber, khi đó $arrJson['OrderNumber'] sẽ văng lỗi logic trầm trọng gây dừng chương trình.
      Nên code như sau:

```
$arrJson = json_decode($json, true);
if (isset($arrJson['OrderNumber'])  && $arrJson['OrderNumber']) {
 // do something
}
```

   4. Sử dụng các hàm array_diff(); in_array(); … (xem thêm tại php.net) để xử lí nhanh với array.
   5. Khi insert cùng lúc nhiều dữ liệu xuống database, thay vì sử dụng vòng foreach và hàm insert thì nên build thành array các Document rồi dùng hàm insertMany để insert xuống, sẽ tăng perfomance cho ứng dụng.
   6. Khi muốn lấy nhiều dữ liệu, thay vì lấy hết dữ liệu, ta phân trang, lấy nhiều lần, mỗi lần 500000, tận dụng skip và limit của query.

III. Cấu trúc 1 Adapter
   1. Cấu hình:
      - Tất cả cấu hình của project nằm trong folder_project/.env
      - Để lấy thông số, sử dụng hàm env('TEN_PARAM');
      Ví dụ:
         Tại file .env: ghi như sau:
         DB_DATABASE=adp_lazada
      Để lấy tên database trong code PHP, ta gọi như sau:

```
$dbname = env('DB_DATABASE' );
```

      - Mỗi Adapter đều là một module con của folder app/ Adapter, do vậy để tạo mới 1 Adapter, vui lòng thêm dòng sau vào file .env để báo cho project là sẽ load code ở đúng module đó:
      ADAPTER=TEN_ADAPTER
      - Sau đó vào folder app/Adapter tạo folder TEN_ADAPTER.
      Ví dụ:
         Tại file .env: ghi như sau:

ADAPTER=Lazada

Sau đó tạo folder Lazada trong folder app/Adapter

2. Cấu trúc thư mục Adapter:

app/
--------- Adapter/
----------/Adapter_Name/
----------/Console
----------FirstCommand.php
–----------SomeCommand.php
----------/Model
----------Dao/
----------LogOrder.php
----------LogOrder.php
----------/Providers
----------ConsoleServiceProvider.php
----------app.php

Còn tất cả các hằng số hay Helper ta sẽ viết trong app/MyLibs.

Chức năng từng file:
- các loại file Command: dùng để chạy cronjob
- các file model và dao: tương tác với database mongodb
- các file Providers: trong adapter chủ yếu dùng để đăng kí các file command
- file app.php: đăng kí sẽ load ServiceProvider nào

Ví dụ: (đừng cố gắng copy code vì định dạng ở trình soạn thảo văn bản sẽ khác các Editor cho code)
Tạo file FirstCommand:

```
namespace App\Adapter\Adapter_Name\Console;

use Illuminate\Console\Command;

class FirstComamnd extends Command
{
	/* khi chạy cronjob sẽ gọi: php artisan some:thing:here */
	protected $signature = ‘some:thing:here’;

	protected $description = ‘mo ta cho command nay’;

	public function handle()
	{
		// do some thing
	}
}
```

Sau đó đăng kí command này cho ServiceProvider bằng cách vào file

ConsoleServiceProvider..php ghi:

```
namespace App\Adater\Adapter_Name\Providers;

use Illuminate\Support\ServiceProvider;
use App\Adapter\Adapter_Name\Console\FirstCommand;

class ConsoleServiceProvider extends ServiceProvider
{
        public function register()
        {
                $this→app→singleton(‘first.command’, function(){
                        return new FirstCommand();
                });
                $this->comamnds(‘first.command’);
        }
}
```

Cấu trúc file app.php:

```
$app = new Laravel\Lumen\Application(
    realpath(__DIR__)
);
$app->configureMonologUsing(function($monolog){
  if (env('APP_DEBUG')) {
    if (!is_writeable(SYS_PATH . '/storage/.')) {
      exit('forlder /storage/. must be allow write');
    }
  }
});




$app→singleton(
        Illuminate\Contracts\Debug\ExceptionHandler::class,
        App\Exceptions\Handler::class
);

$app→singleton(
        Illuminate\Contracts\Console\Kernel::class,
        App\Console\Kernel::class
);
$app→register(App\Providers\AppServiceProvider::class);
$app→register(
 App\Adapter\Adapter_Name\Providers\ConsoleServiceProvider::class
);

return $app;
```

Cấu trúc 1 file Model:

```
namespace App\Adapter\Adapter_Name\Model;

use App\MyLibs\Singleton;

class LogOrder extends Singleton
{
        private $_dao;

        protected function __construct()
        {
                parent::__construct();
                $this->_dao = new Dao\LogOrder();
        }
}
```

Cấu trúc 1 file Dao:

```
namespace App\Adapter\Adapter_Name\Model\Dao;

use App\MyLibs\DBCollection\Log\OrderBaseCollection;

class LogOrder extends OrderBaseColletion
{
}
```

Trong đó class OrderBaseCollection được tạo tự động khi chạy lệnh:

php artisan schema:builder

ở giao diện terminal.

Tham khảo cách đăng kí các BaseCollection tại folder app/MyLibs/Schema/

3. Tổng quan về Adapter:
   a. Chức năng chính:
      - Luôn chuyển dữ liệu và format dữ liệu để đồng bộ Channel và SCP
      - Dữ liệu được xử lí bất đồng bộ, chạy theo cronjob.
   b. Một số chú ý:
      - Mỗi lần lấy dữ liệu từ Channel hoặc SCP nên log lại xuống database. Sau khi gọi API để luân chuyển thì cập nhật dữ liệu đã chuyển rồi, đề phòng khi luân chuyển không được, thì sẽ được luân chuyển từ đợt sau.
      Ví dụ:
      - Lazada có 10 đơn hàng, Adapter từ Lazada sẽ gọi API đổ dữ liệu từ Lazada vào SCP. Giả sử có gì đó sai sót, SCP chỉ nhận được 8 đơn hàng. Còn 2 đơn hàng bị lỗi, sẽ được Adapter lưu lại vào database.
      - 10 phút sau, Lazada báo có 20 đơn hàng, khi đó Adapter sẽ lấy 2 đơn hàng bị lỗi gửi chung với 20 đơn hàng đợt này, tổng cộng là 22 đơn xuống SCP.