



面對現存資料庫 反轉 Laravel Migration 及 Seeder 的解決方案

范聖佑 Shengyou Fan

 OSC 源創會第 64 期 (杭州)

2017/06/24



```
{  
  "name": "shengyou/self-introduction",  
  "description": "自我介紹",  
  "authors": [  
    {  
      "name": "范聖佑 (Shengyou Fan)",  
      "email": "shengyoufan@gmail.com",  
      "homepage": "http://www.shengyoufan.com",  
      "role": ["研究員", "傳教士"]  
    }  
  ],  
  "support": {  
    "facebook": "http://fb.me/shengyoufan",  
    "twitter": "@shengyou",  
    "wechat": "@shengyoufan"  
  }  
}
```





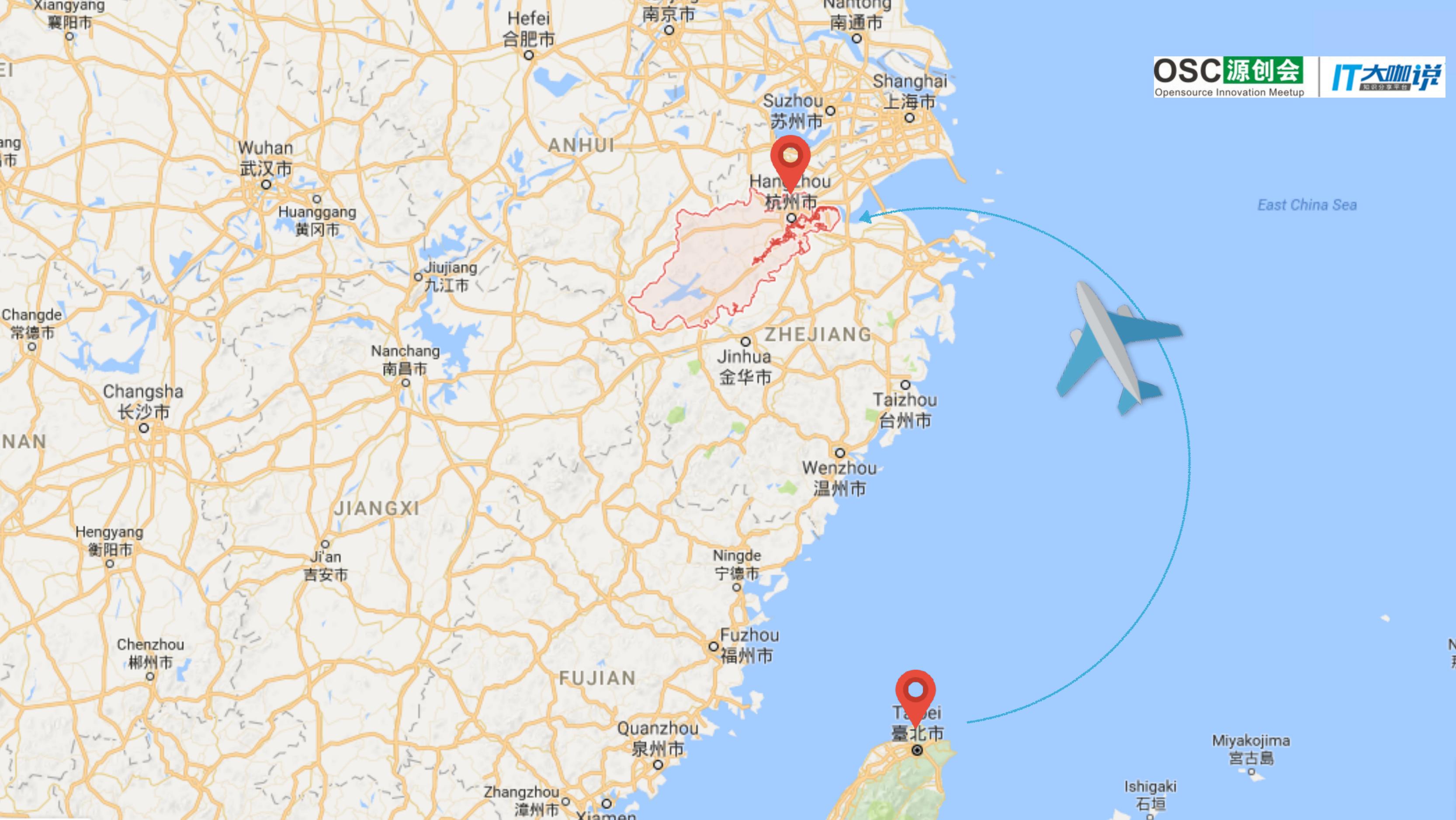
Yangmingshan
National Park
陽明山
國家公園

Keelung City
基隆市

Taipei
臺北市

New
Taipei City

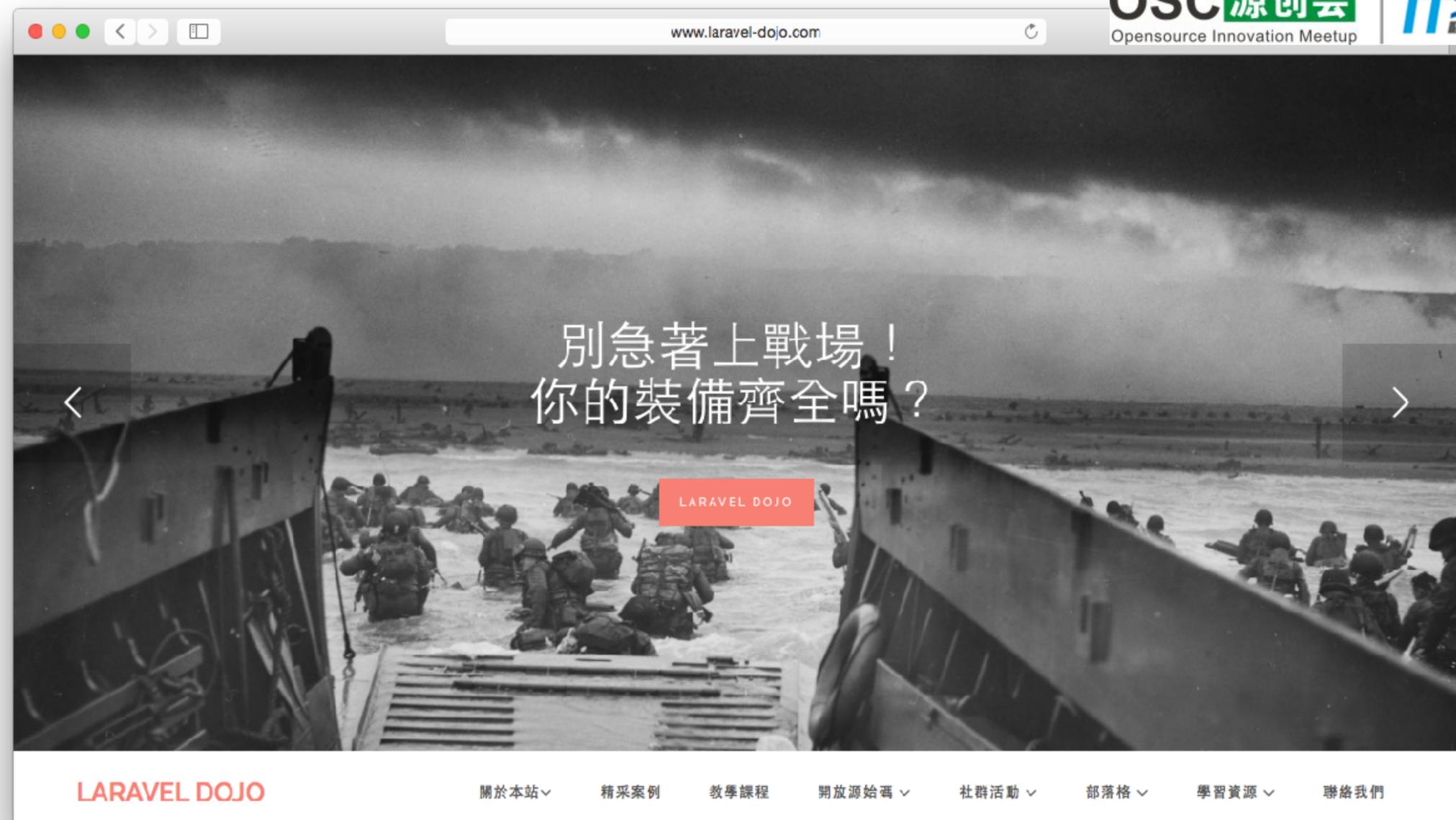
Taoyuan City



East China Sea

Miyakojima
宫古島

Ishigaki
石垣



Laravel Dojo

致力於提供最好的 PHP/Laravel 教育訓練

<http://www.laravel-dojo.com>

LaravelConf Taiwan 2017

<https://laravelconf.tw/>



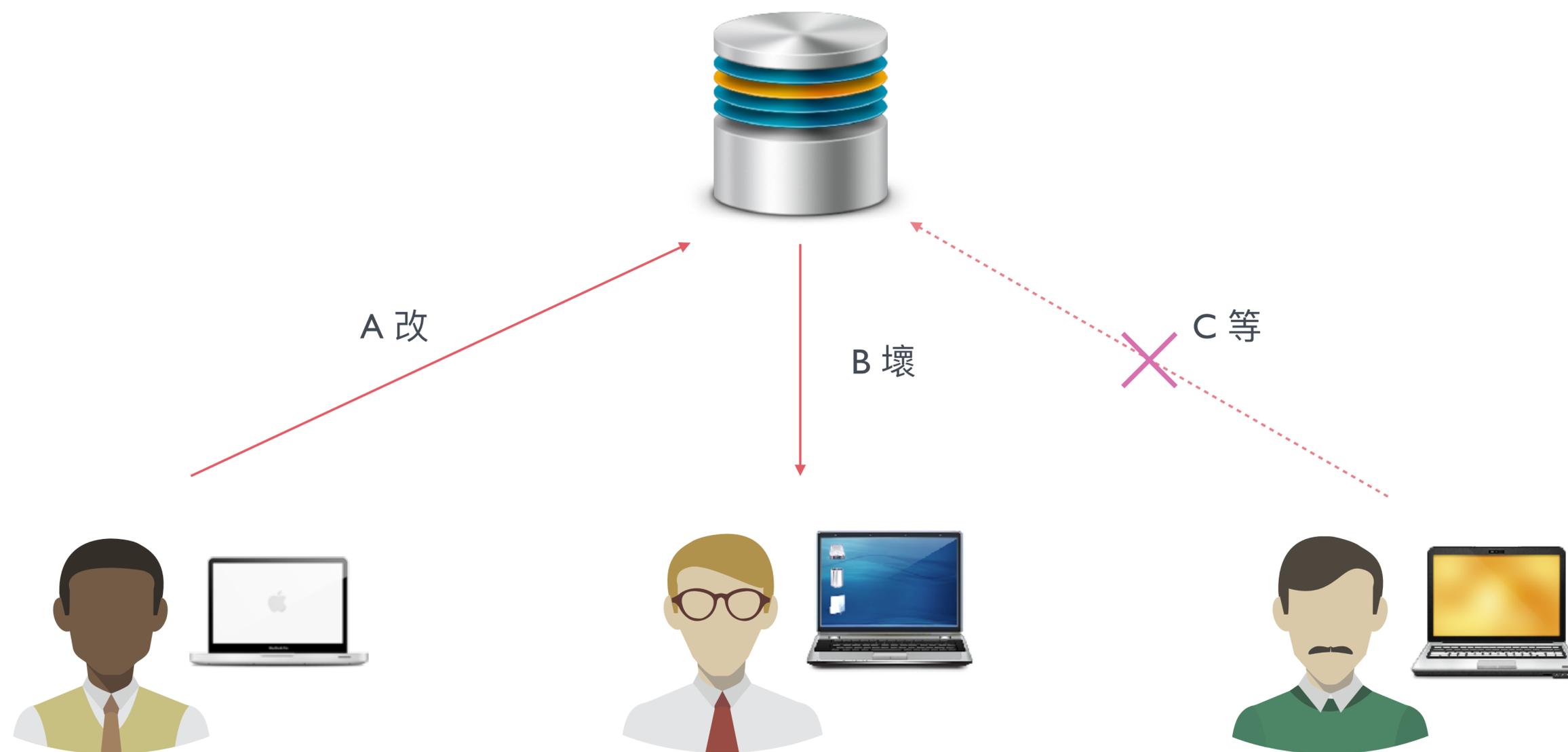
主題大綱

- 團隊協同開發時，資料庫同步的需求
- Laravel Migration 及 Seeding 機制
- 殘酷的現實情境
- 反轉 Migration & Seeder 的套件
- 其他可行方案



多人協同開發時 資料庫版本管理的需求

共用資料庫



獨立資料庫

結構與內容需要同步



更新資料庫



資料庫也需要版本管理



Laravel 的解決方案

導入 Migration 及 Seeding 機制

Migrate 流程

- 先用 `artisan` 產生 migration 檔
- 撰寫 migration 檔的內容
- 使用 `artisan` 執行 `migrate` 指令
- 測試 `rollback`

```
$ [php] artisan make:migration ...
```

使用 artisan 建立一個 migration



```
// 一個 migration 的基本結構
class {Name} extends Migration
{
    public function up()
    {
        // ...
    }

    public function down()
    {
        // ...
    }
}
```

```
// 建立資料表
class {Name} extends Migration
{
    public function up()
    {
        Schema::create('{name}', function (Blueprint $table) {
            $table->increments('id');
            $table->timestamps();
        });
    }

    public function down()
    {
        Schema::dropIfExists('{name}');
    }
}
```

```
// 修改資料表
class {Name} extends Migration
{
    public function up()
    {
        Schema::table('{name}', function (Blueprint $table) {
            $table->{type}('column');
            $table->{type}('column')->change();
        });
    }

    public function down()
    {
        Schema::table('{name}', function (Blueprint $table) {
            $table->dropColumn('column');
            $table->{type}('column')->change();
        });
    }
}
```

```
$ [php] artisan migrate
```

使用 artisan 執行 migrate 指令

```
$ [php] artisan migrate:rollback
```

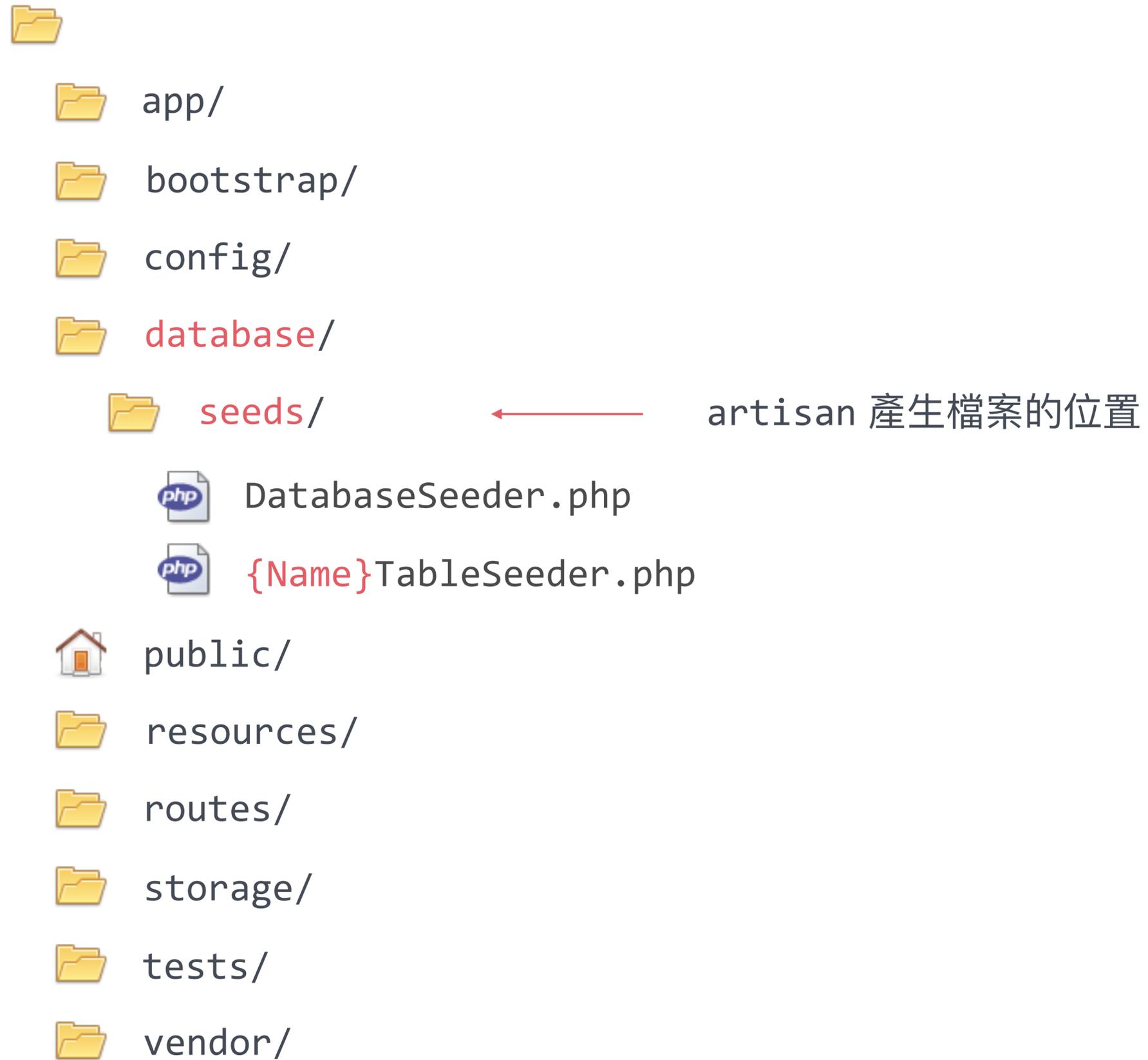
還原資料庫至上一個版本

Seeding 流程

- 用 `artisan` 產生 Seeder 檔
- 撰寫 Seeder 檔的內容
- 設定 `DatabaseSeeder.php` 內的執行序
- 使用 `artisan` 執行 `db:seed` 指令

```
$ [php] artisan make:seeder ...
```

使用 artisan 建立一個 seeder



```
// 一個 seeder 的基本結構
class {Name} extends Seeder
{
    public function run()
    {
        // ...
    }
}
```

```
// 讓 Seeder 更好用
class {Name} extends Seeder
{
    public function run()
    {
        DB::table('{name}')->truncate();

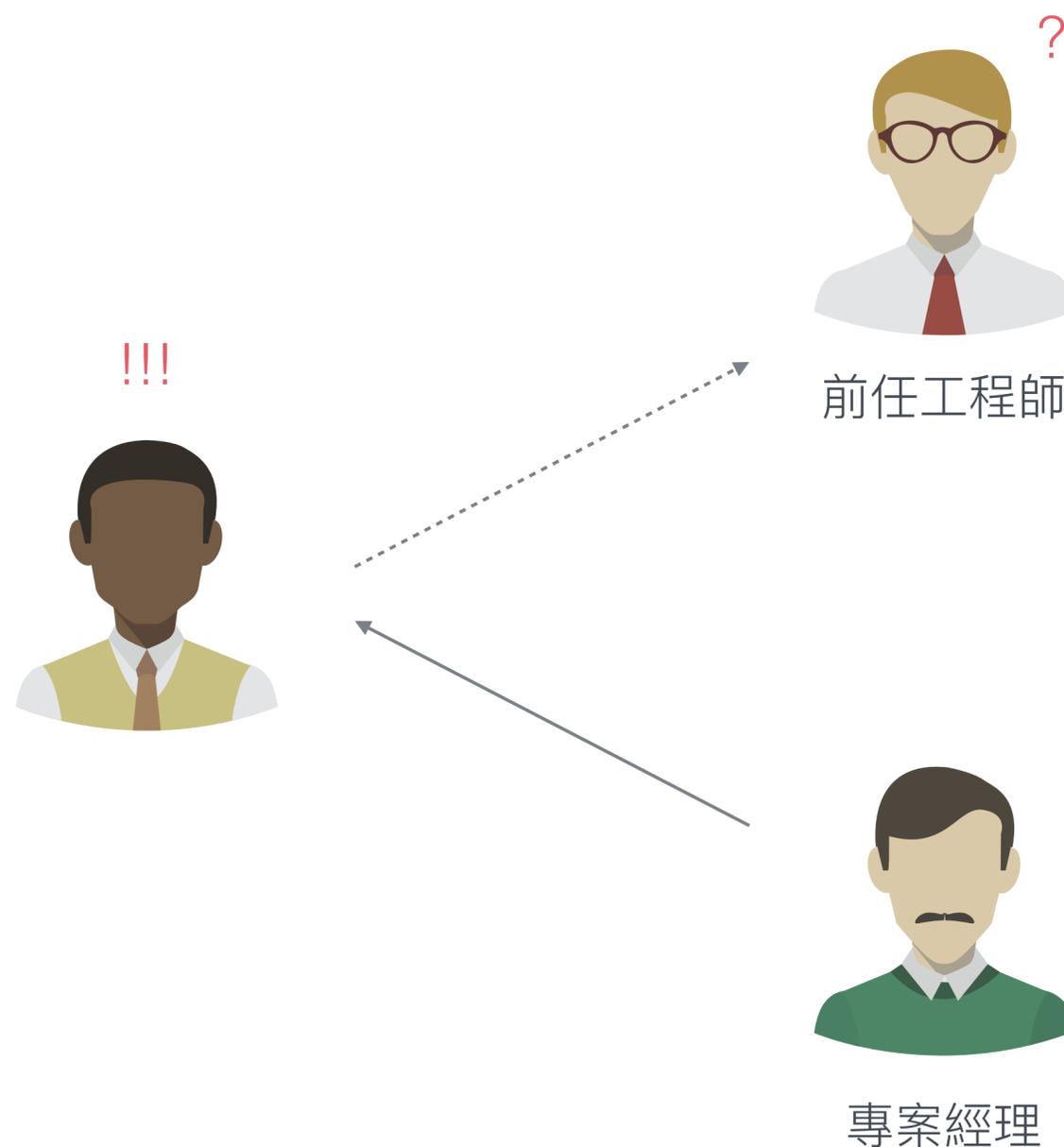
        $faker = Faker::create('{locale}');

        foreach(...) {
            Model::create([
                // ...
                'created_at' => Carbon::now()->subDays({number}),
                // ...
            ]);
        }
    }
}
```

```
$ [php] artisan db:seed
```

使用 artisan 執行資料 seeding

(殘酷的) 現實情境



Migration 是什麼？可以吃嗎？

資料庫已經用很久了喔~別亂改...

資料表和欄位名稱都是前人取的...

新需求 (!)

現在要把資料庫搬到...

(̄ ̄ ̄) ̄

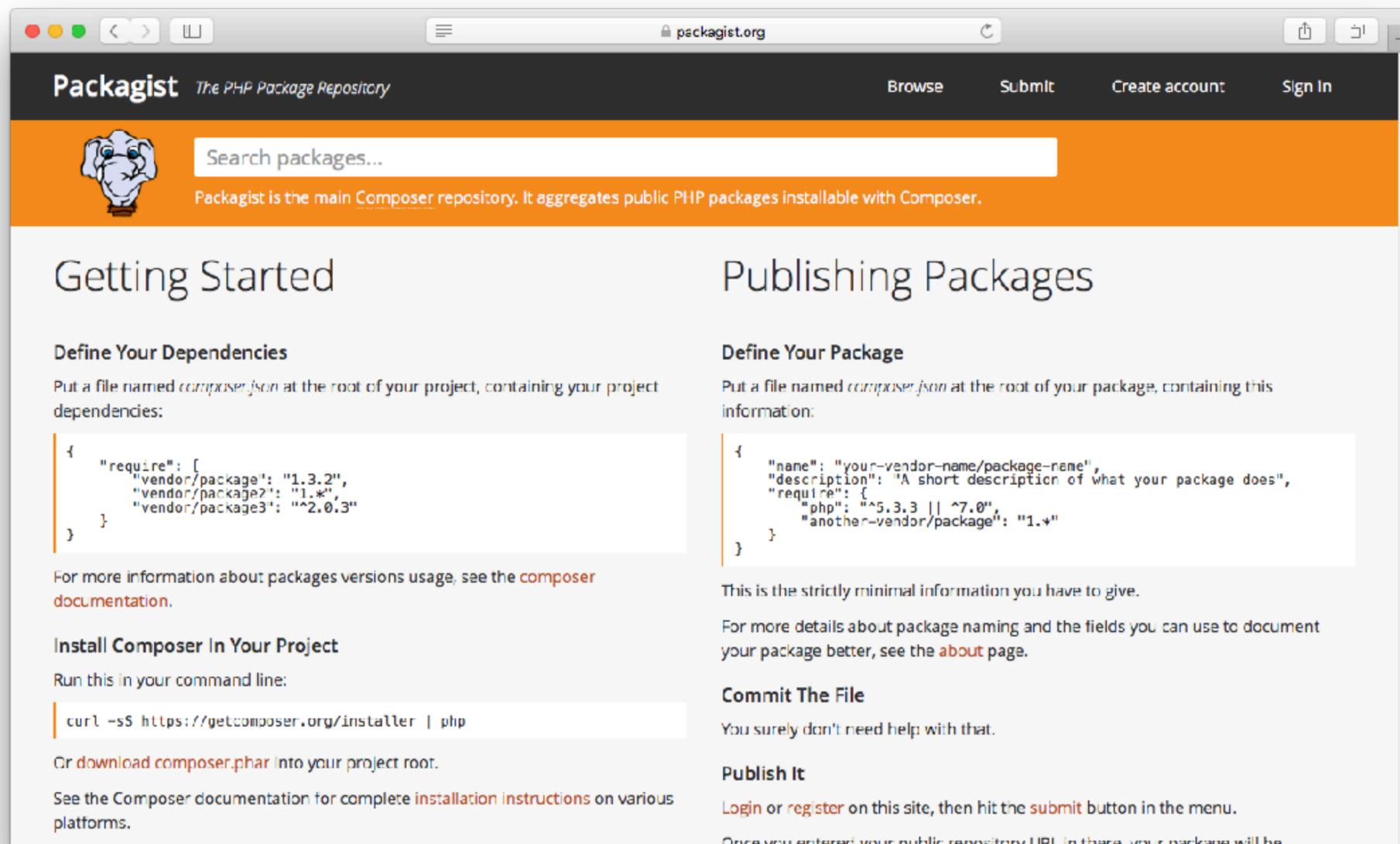
翻桌！



讓套件生態系來幫你！

從現有資料庫反轉 Migration 及 Seeding

Composer 生態系



- ★ 來源：<https://packagist.org/>
- ★ 來源：<https://getcomposer.org/>

安裝套件

- 安裝 xethron/migrations-generator 套件

```
$ composer require --dev xethron/migrations-generator
```

註冊 ServiceProvider

- 設定 app/Providers/AppServiceProvider.php

```
// 在非 production 環境下註冊 ServiceProvider
public function register()
{
    // ...
    if (! $this->app->environment('production')) {
        $this->app->register(Generators...::class);
        $this->app->register(Migrations...::class);
    }
}
```

```
$ [php] artisan migrate:generate
```

反轉現有資料庫結構成 migration

```
$ [php] artisan migrate:generate table1,table2,table3,table4,table5  
$ [php] artisan migrate:generate --ignore="table3,table4,table5"
```

安裝套件

- 安裝 orangehill/iseed 套件

```
$ composer require --dev orangehill/iseed
```

註冊 ServiceProvider

- 設定 app/Providers/AppServiceProvider.php

```
// 在非 production 環境下註冊 ServiceProvider
public function register()
{
    // ...
    if (! $this->app->environment('production')) {
        // ...
        $this->app->register(Iseed...::class);
    }
}
```

```
$ [php] artisan iseed {table}
```

反轉現有資料庫內容成 seeder

```
$ [php] artisan iseed table1,table2,table3,table4,table5  
$ [php] artisan iseed table1 --force --clean
```

搬移至其他資料庫

- 在 config/database.php 裡設定連線來源

```
'connections' => [  
    'sqlite' => [  
        // ...  
    ],  
  
    'mysql' => [  
        // ...  
    ],  
  
    'pgsql' => [  
        // ...  
    ],  
],
```

修改環境設定

- 修改 .env 裡的資料庫連線設定

```
DB_CONNECTION=  
DB_HOST=  
DB_PORT=  
DB_DATABASE=  
DB_USERNAME=  
DB_PASSWORD=
```

- 執行 migrate 及 seeding 指令

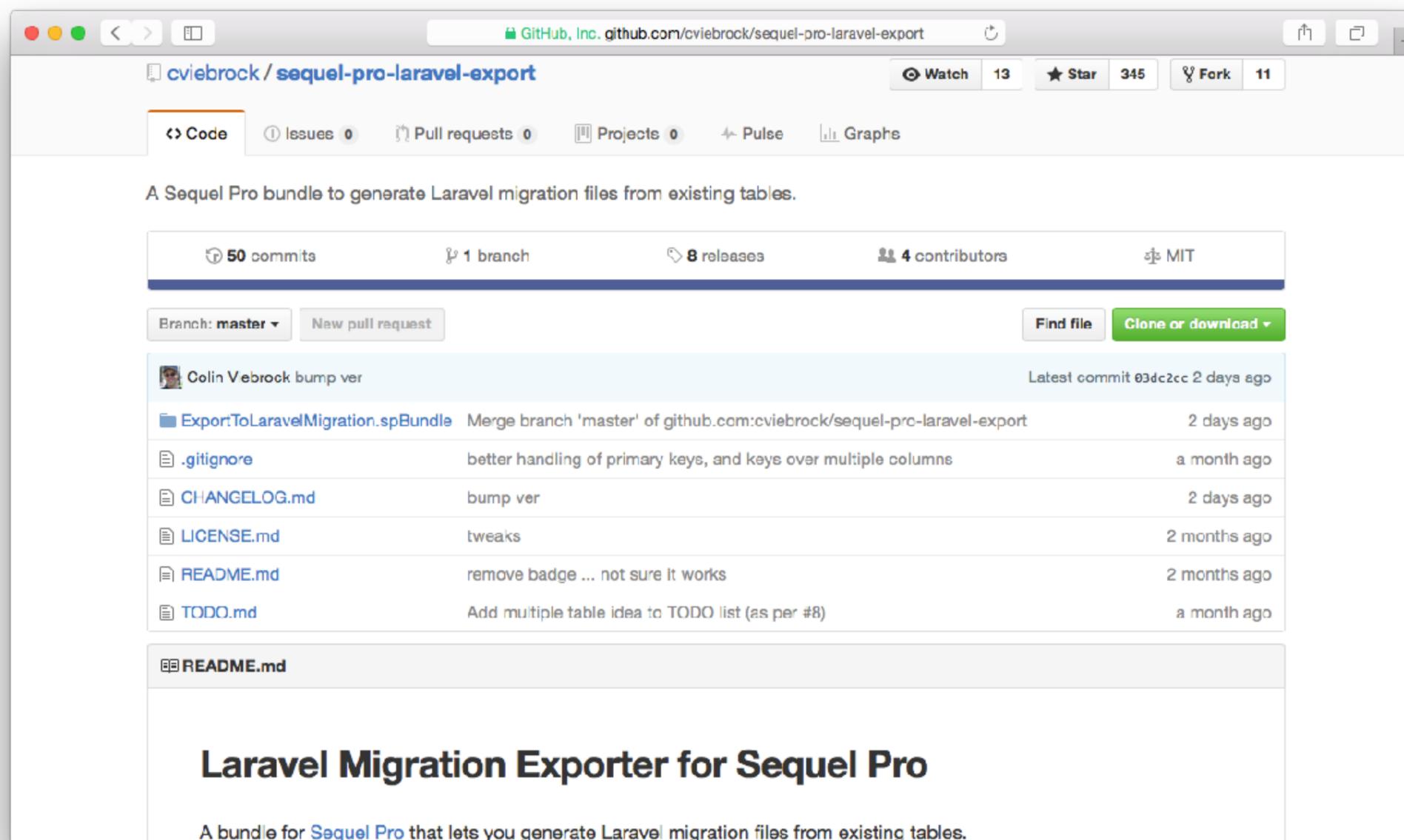
```
$ [php] artisan migrate  
$ [php] artisan db:seed
```

其他可行方案

若不想使用套件來處理的話...

使用 Sequel Pro

<https://github.com/cviebrock/sequel-pro-laravel-export>



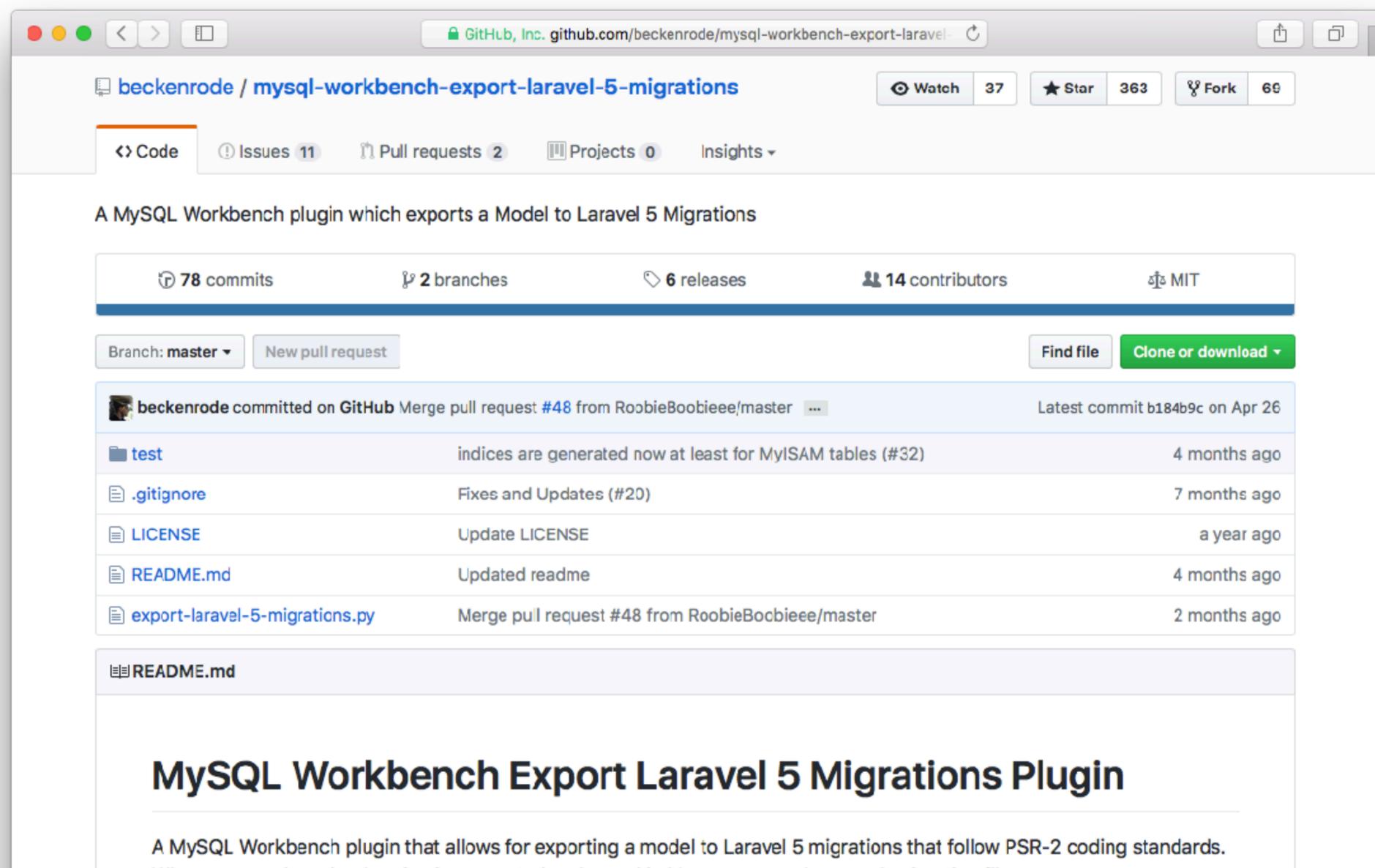
The screenshot shows the Sequel Pro application interface. A context menu is open over a table, with the 'Export To Laravel Migration' option selected. The table has columns for 'id', 'content', 'post_date', 'cat_id', and 'create_date'. The 'TABLE INFORMATION' panel at the bottom left shows details for the 'news' table, including engine (InnoDB), rows (67), size (112.0 KiB), encoding (utf8), and auto_increment (91). A notification window at the bottom right displays the path where the migration file was saved.

id	content	post_date	cat_id	create_date
2				
4				
5				
10				
16				
17				
21				
22				
24				
25				
26				
28				
31				
32				
33				
34				
35				
36				
37				
38				
39				
40				
42				
43				
44				
46				

Migration saved to ~/Desktop/SequelProLaravelExport/2017_06_18_001704_create_news_table.php.

使用 MySQL Workbench

<https://github.com/beckenrode/mysql-workbench-export-laravel-5-migrations>



- Configuration
- Utilities
- Catalog
- Objects
- Start Shell for MySQL Utilities

- Export Laravel 5 Migration
- Dump All Table Columns
- Copy Table List to Clipboard
- Give a Prefix to All Tables in Catalog
- Change the Storage Engine of All Tables

MySQL Model x

Description

No Selection

Physical Schemas

Tables (1 item)

- Add Table
- user

Views (0 items)

- Add View

user - Table

Name: user Schema: mydb

Collation: Schema Default Engine:

Comments:

Column	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	G	Default / Ex...
username	VARCHAR(16)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
email	VARCHAR(255)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
password	VARCHAR(32)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
create_time	TIMESTAMP	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	CURRENT_TI...
<click to edit>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Columns Indexes Foreign Keys Triggers Partitioning Options Inserts Privileges

SQL Editor closed

timestamps
create_time, update_time

user
username, email, password, crea...

category
category_id, name

Plugin Manager

Plugins

- Export Laravel 5 Migration

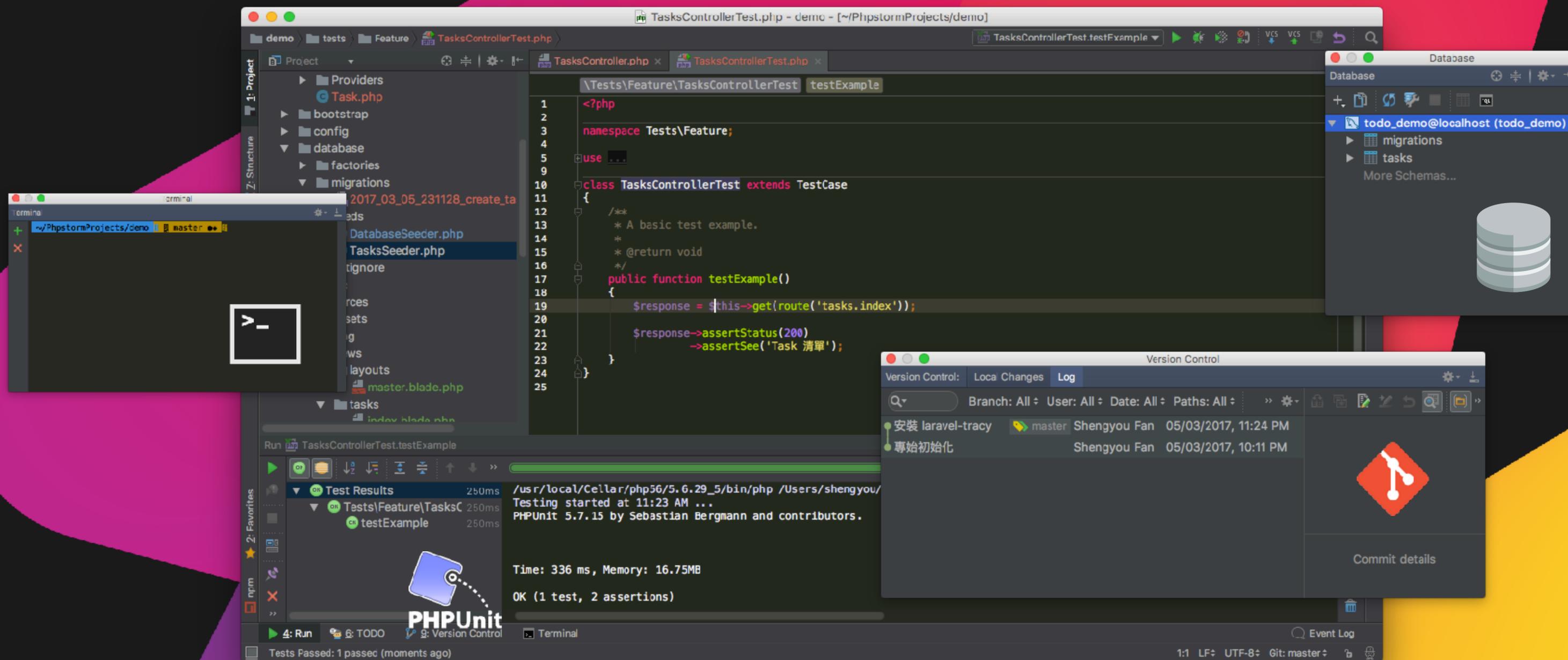
Export Laravel 5 Migration

 Name: wb.util.generateLaravel5Migration
version 0.1.5 by Brandon Eckenrode

Enable Plugin Uninstall

Show plugin details Close

PhpStorm 整合開發環境



★ JetBrains 針對學生、教育單位及開放原始碼專案的作者有提供免費授權的申請，符合資格的話別錯過了！



The Qwan 得寬科技

自 2013 年導入

100 % 採用 Laravel 開發的團隊

JetBrains 教育訓練夥伴
(在台灣首間也是唯一一間)





Q & A

感謝聆聽 · 歡迎交流