

PHP Faker 활용

빠르고 간편하게 의미 있는 테스트 데이터 만들기

정광섭

- lesstif@gmail.com
- <http://lesstif.com>

목차

- ▶ 테스트 데이터의 필요성
- ▶ PHP Faker 란?
- ▶ 사용 예제
- ▶ Laravel 과 연동

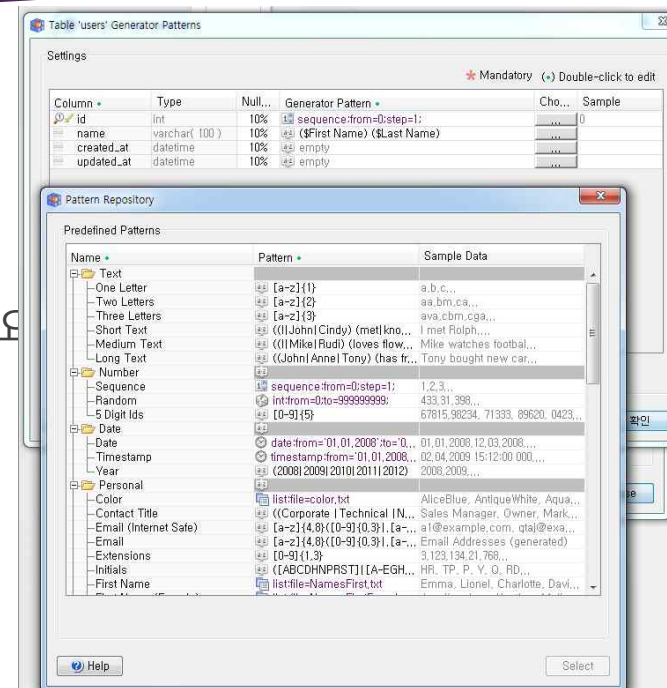
의미 있는 테스트 데이터의 필요성 #1

- ◆ 설계와 개발 업무를 진행하면서 여러 단계에서 테스트 데이터가 필요
 - ▶ 설계한 DB 스키마 검증 및 쿼리 코드의 정상 동작 여부 확인
 - ▶ API의 동작 여부 확인
 - ▶ View를 만들면서 데이터에 맞게 뷰 조정(레이아웃, 그리드, 페이징 등)
 - ▶ 단위 테스트, 성능 및 스트레스 테스트
- ▶ 위 요구사항을 만족하는 의미 있는 테스트 데이터(Real Test data)가 필요
- ▶ 필요하지만 이런 데이터를 만들기 위해 시간을 투자할 수는 없음

의미있는 테스트 데이터의 필요성 #2

◆ 테스트 데이터 만드는 법

- ▶ 수작업(SQL 직접 입력등)
- ▶ 운영 데이터를 테스트 환경으로 복제
 - ▶ 개인 정보가 포함되었을 경우 정보통신망법, 개인정보보호법 준수 필요
 - ▶ 신규 개발 서비스일 경우 운영 데이터가 없음
- ▶ 툴을 사용하여 테스트 데이터 생성
 - ▶ Ex: DBSchema 의 random data generation 기능



PHP Faker 란?

- ▶ 의미있는 Faker 데이터를 만들어 주는 패키지(주소, 이름, 문장, 카드 번호등)
- ▶ Github 6,500 Star 에 빛나는 패키지 (<https://github.com/fzaninotto/Faker>)
- ▶ Perl 의 Data::Faker, ruby 의 Faker 로부터 강한 영향을 받음
 - ▶ python 의 Faker(<https://github.com/joke2k/faker>) 는 PHP Faker 의 영향을 받아 개발
 - ▶ 언어/프레임워크 마다 별도의 Faker 패키지가 있고 모두가 많은 별점을 받았다는 것은 테스트 데이터 생성이 필요한 경우가 많았음을 의미함.

사용 예제 #1

▶ 설치

```
composer require fzaninotto/faker
```

▶ 사용

- ▶ `$faker->property` 를 사용하면 `__get` magic method 에 의해 `Faker\Generator->format($property)` 실행
- ▶ Address, Name, Company, DateTime, Text 등 특정 영역의 데이터를 생성하는 Provider 클래스에서 실제 데이터 생성(`src/Faker/Provider/`)

```
<?php
```

```
require_once 'vendor/autoload.php';
```

```
$faker = Faker\Factory::create();
```

```
// generate data by accessing properties
```

```
echo $faker->name; // 'Lucy Cechtelaar';
```

```
echo $faker->address; // "426 Jordy Lodge Cartwrightshire, SC 88120-6700"
```

```
echo $faker->text;
```

사용 예제 #2 – 사용자 정보 생성

```
public function testUserCreate()
{
    $faker = Faker\Factory::create();

    $user = [
        'id' => $faker->randomNumber($nbDigits = NULL), //numberBetween($min = 1000, $max = 9000)
        'name' => $faker->name($gender = 'female'),
        'country' => $faker->country,
        'address' => $faker->address,
        'phoneNumber' => $faker->phoneNumber,
        'company' => $faker->company,
        'birthDay' => $faker->dateTimeBetween('-50 years', '-20 years'),
        'email' => $faker->email, // safeEmail, freeEmail, etc..
        'homePage' => $faker->url,
        'creditCardType' => $faker->creditCardType,
        'creditCardNumber' => $faker->creditCardNumber,
    ];

    dump($user);
}
```

사용 예제 #3 – 첨부 파일 정보 생성

```
public function testAttachmentCreate()
{
    $faker = Faker\Factory::create();

    $attachment = [
        'id' => $faker->randomNumber($nbDigits = NULL),
        'user_id' => $faker->numberBetween($min = 1000, $max = 9000),
        'mimeType' => $faker->mimeType,
        'size' => (1024 * $faker->numberBetween($min = 12345678, $max = 987654321)),
        'path' => $faker->file($srcDir = '.' , $destDir = 'storage', $fullPath = false),
    ];

    dump($attachment);
}
```


라라벨과 연동 #1 - Model Factory 선언

- ▶ 라라벨 5.1 부터 Faker 와 연동한 Model Factories 기능이 추가됨 (<http://laravel.com/docs/5.1/testing#model-factories>)
- ▶ 모델 팩터리 지정 - **database/factories/ModelFactory.php**
- ▶ **\$factory->define** 메소드에 사용할 Model 클래스를 전달하고 클로저에 \$faker 데이터 지정

```
$factory->define(App\Author::class, function ($faker) {  
    return [  
        'name' => $faker->name,  
        'email' => $faker->email,  
        'country' => $faker->country,  
        'birthDay' => $faker->dateTimeBetween('-50 years', '-20 years'),  
        'password' => str_random(10),  
        'remember_token' => str_random(10),  
    ];  
});
```

라라벨과 연동 #2 - Faker data 생성

- ▶ factory() 헬퍼의 make() 메소드를 사용하여 Faker 데이터 생성
- ▶ 첫 번째 파라미터 - 모델 클래스
- ▶ 두 번째 파라미터(옵션) - 생성할 갯수

```
>>> factory('App\Author', 1)->make();  
=> <App\User #00000000439a1a48000000001f1fd334> {  
    name: "Darrin Farrell",  
    email: "Seth37@Schaden.com"  
}
```

라라벨과 연동 #3 - DB 에 입력

- ▶ make() 메소드는 DB 에 입력하지 않고 모델만 생성
- ▶ DB 에 입력하려면 create() 메소드 사용

```
>>> factory('App\Author', 10)->create();  
=> <App\User #00000000439a1a480000000001f1fd334> {  
    name: "Darrin Farrell",  
    email: "Seth37@Schaden.com"  
}
```

라라벨과 연동 #4 - Model Relation

- ▶ Factory Model 생성후 each() 로 자식 모델 생성

```
$authors = factory(App\Author::class, 3) ->create()
    ->each(function($u) {
        $u->books()->save(
            factory(App\Book::class, 10)->make());
    });
```

라라벨과 연동 #5 - Model Relation

- ▶ 자식 factory 에서 부모의 id 값을 가져와서 범위 지정

```
$factory->define(App\Book::class, function ($faker) use($factory){
    // 최대값과 최소값 가져오기
    $max = App\Author::max('id');
    $min = App\Author::min('id');
    return [
        'name' => $faker->text,
        'isbn' => $faker->isbn13,
        // Author id 참조
        'author_id' => $faker->numberBetween($min, $max),
        'created_at' => $faker->dateTimeBetween($startDate = '-2 years', $endDate = '-1 years'),
        'updated_at' => $faker->dateTimeBetween($startDate = '-1 years', $endDate = 'now'),
    ];
});
```

참고 자료 & QnA

- ▶ Faker 홈 페이지 - <https://github.com/fzaninotto/Faker>
- ▶ Laravel Model Factory - <http://laravel.com/docs/5.1/testing#model-factories>
- ▶ 더 자세한 내용은 <http://lesstif.com/x/7QKOAQ> 참고

▶ Q&A