**TDD trong Laravel**

**Test Driven Development (TDD) là một phương pháp phát triển phần mềm, thay vì viết code sau đó viết test case để kiểm thử (Test Last Development) . Ta làm theo hướng ngược lại, viết các test case để kiểm thử trước , sau đó viết code để thỏa mãn các test case đó. (Test First Development)**

1. **Cách thực hiện TDD**
2. **Viết một test case ( màu đỏ )**

*Viết code cần và (vừa)đủ để làm sao pass được test case vừa viết*

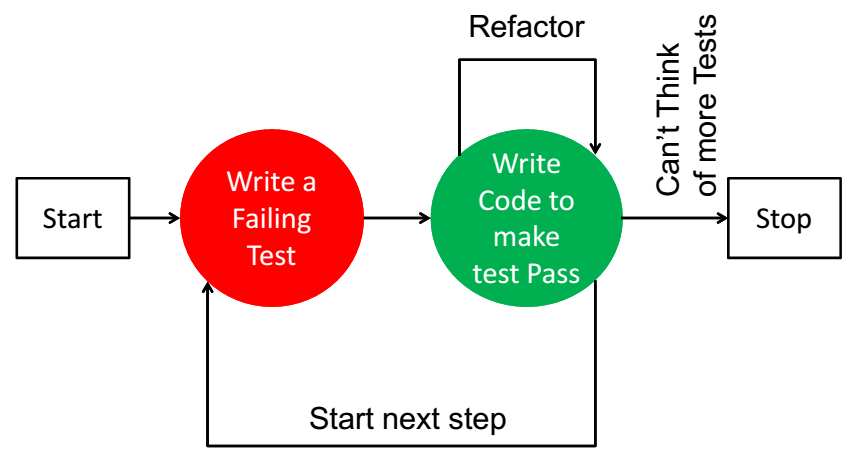
*Nghĩ về một tính năng và viết test case cho nó, vì tính năng này chưa được implement, test case này chắc chắn sẽ fail*

1. **Sửa code cho test pass ( màu xanh )**

*Viết code cần và (vừa)đủ để làm sao pass được test case vừa viết*

1. **Loại bỏ code dư thừa, cải thiện code (Refactor)**

*Vì ở bước 2 ta chỉ viết cho (vừa)đủ để pass test case , sau khi implement đầy đủ các tính năng thì giờ đã đến lúc để nghĩ đến việc refactoring code làm sao cho dễ đọc, dễ hiểu hoặc làm sao cho nó có hiệu năng tốt hơn*

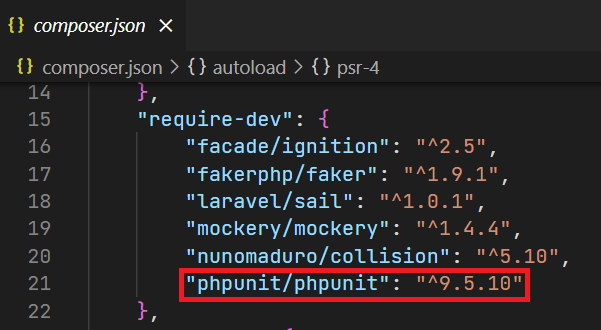
[](file:///C:\Users\Tuan\AppData\Local\Temp\Rar$EXa5248.47518\TDD%20trong%20Laravel%20adbeef81a4bf4a308b0e4cfd8c1b2850\1.png)

***Thực hiện quy trình trên cho đến khi implement và pass được tất cả các requirements được đặt ra ⇒ Code được 100% test coverage***

**Xây dựng một website Todolist Laravel đơn giản áp dụng TDD**

**Ở đây ta sẽ xây dựng một app Todolist đơn giản áp dụng quy trình TDD**

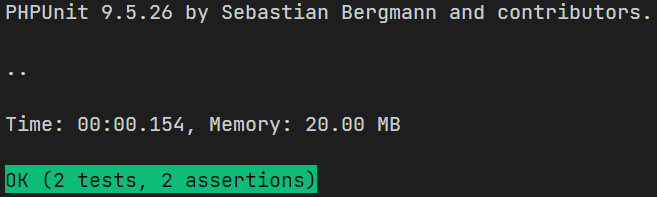
Đối với một project laravel mới được tạo, laravel đã được mặc định kèm sẵn plugin để viết unit test, plugin này có tên là ***phpunit***

[](file:///C:\Users\Tuan\AppData\Local\Temp\Rar$EXa5248.47518\TDD%20trong%20Laravel%20adbeef81a4bf4a308b0e4cfd8c1b2850\Untitled.png)

**Cách sử dụng phpunit để chạy unit test ( trên nền CLI )**

Ta sử dụng lệnh: ***vendor/bin/phpunit***

Ta được kết quả như sau:

[](file:///C:\Users\Tuan\AppData\Local\Temp\Rar$EXa5248.47518\TDD%20trong%20Laravel%20adbeef81a4bf4a308b0e4cfd8c1b2850\Untitled%201.png)

Sở dĩ có 2 tests ở đây là vì trong template của laravel có sẵn 2 file test làm ví dụ, 2 file này nằm lần lượt ở

***tests\Unit\ExampleTest.php***

***tests\Feature\ExampleTest.php***

**Xóa cả hai file này bởi vì chúng ta không cần đến chúng**

Để chạy một test unit cụ thể , ta sử dụng thêm thuộc tính ***—fliter*** kèm với ***class name*** hoặc ***method name***, ví dụ:

***vendor/bin/phpunit --filter ExampleTest***

***vendor/bin/phpunit --filter test\_example***

Ở đây ta sẽ dùng laravel viết một ứng dụng Todolist đơn giản (thêm, xóa đã làm, set đã làm) nhưng theo hướng TDD

Tạo một file Unit Test mới bằng lệnh : **php artisan make:test TodoTest**

Giờ ta có file ***tests\Feature\TodoTest.php***

Giả sử requirement yêu cầu homepage của app todo của chúng ta nằm ở route ***todoapp*** , hay ***http://127.0.0.1:8000/todoapp***

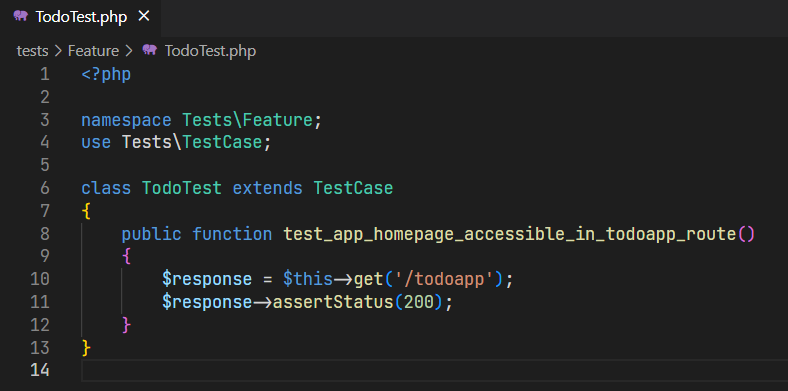
Thay vì lao đầu vào viết code tạo route luôn thì thay vào đó ta sẽ viết test case trước

**Có 2 cách để viết function test**

1. **Thêm tiền tố test\_ vào trước tên hàm**

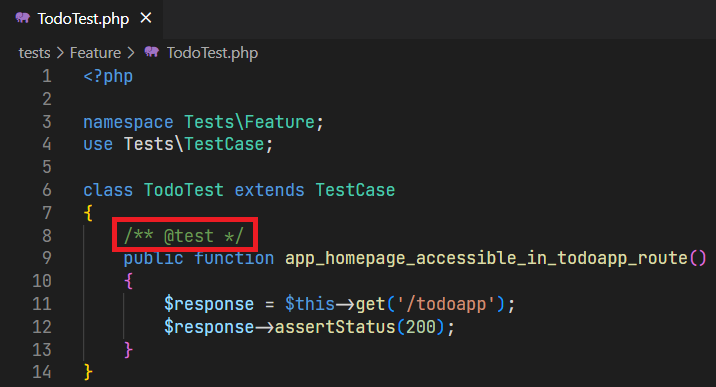
Ví dụ hàm test của mình đặt tên là ***app\_homepage\_accessible\_in\_todoapp\_route*** thì phải thêm tiền tố ***test\_*** vào phía trước

Ta có***: test\_app\_homepage\_accessible\_in\_todoapp\_route***

[](file:///C:\Users\Tuan\AppData\Local\Temp\Rar$EXa5248.47518\TDD%20trong%20Laravel%20adbeef81a4bf4a308b0e4cfd8c1b2850\Untitled%202.png)

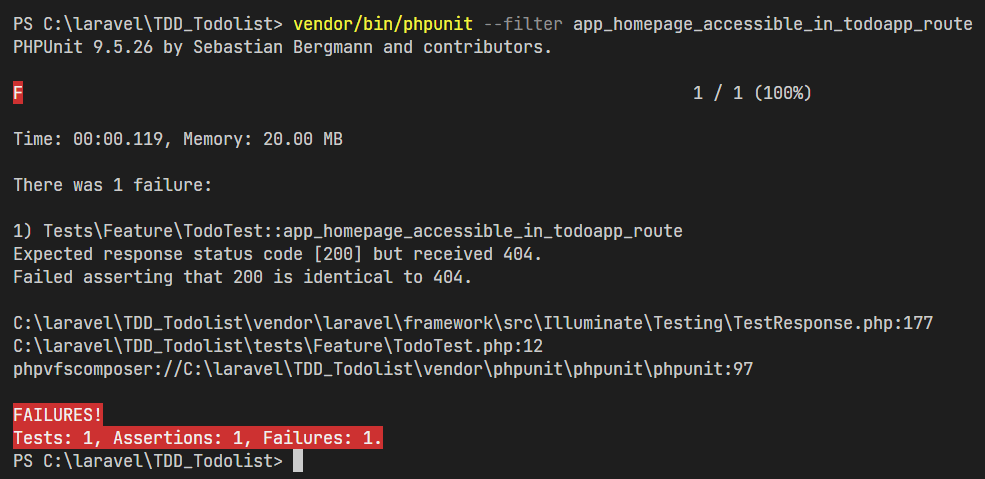
1. **Dùng test annotation**

Với test annotation thì hàm test của mình sẽ không cần tiền tố ***test\_*** nữa nhưng ta phải để test annotation: ***/\*\* @test \*/*** lên trên đầu hàm

[](file:///C:\Users\Tuan\AppData\Local\Temp\Rar$EXa5248.47518\TDD%20trong%20Laravel%20adbeef81a4bf4a308b0e4cfd8c1b2850\Untitled%203.png)

**Bắt buộc phải dùng 1 trong 2 cách trên , nếu không phpunit sẽ báo *No tests found***

Ta chạy test unit vừa mới viết trên

[](file:///C:\Users\Tuan\AppData\Local\Temp\Rar$EXa5248.47518\TDD%20trong%20Laravel%20adbeef81a4bf4a308b0e4cfd8c1b2850\Untitled%204.png)

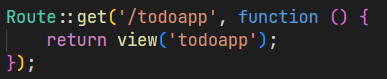
Test case ở trên kiểm tra xem route ***todoapp*** có truy cập được không , nếu truy cập được thì có http status trả về là 200

Ở đây phpunit báo lỗi test failed , ***cơ mà đây chính xác những gì ta hướng đến khi viết theo phương pháp TDD*** (bước 1 của TDD như đã giới thiệu ở trên)

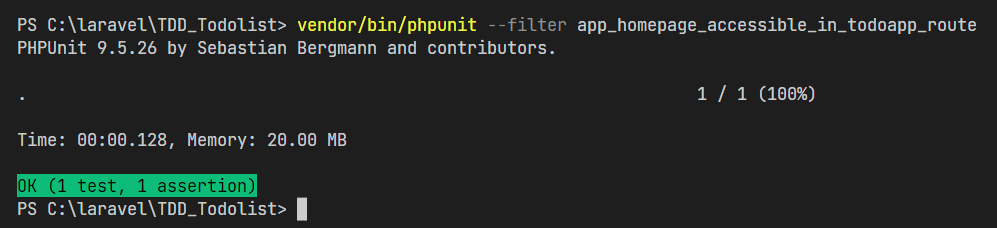
Giờ chúng ta chuyển sang bước 2, viết code để thõa mãn test case ta vừa viết

**Tạo file resources\views\todoapp.blade.php**

**Thêm route vào routes\web.php**

[](file:///C:\Users\Tuan\AppData\Local\Temp\Rar$EXa5248.47518\TDD%20trong%20Laravel%20adbeef81a4bf4a308b0e4cfd8c1b2850\Untitled%205.png)

Chạy lại test case, ta thấy test case đã passed, hoàn thành được mục tiêu test first (**Test Driven Development - TDD**)

[](file:///C:\Users\Tuan\AppData\Local\Temp\Rar$EXa5248.47518\TDD%20trong%20Laravel%20adbeef81a4bf4a308b0e4cfd8c1b2850\Untitled%206.png)

**TDD có một mô hình test 3 pha (3-phase pattern) gồm:**

**Arrange** ⇒ là function khởi tạo môi trường cần thiết cho quá trình test

**Act** ⇒ là khi Arrange đã hoàn thành xong công việc , Act chứa những đoạn code để thực hiện bài test

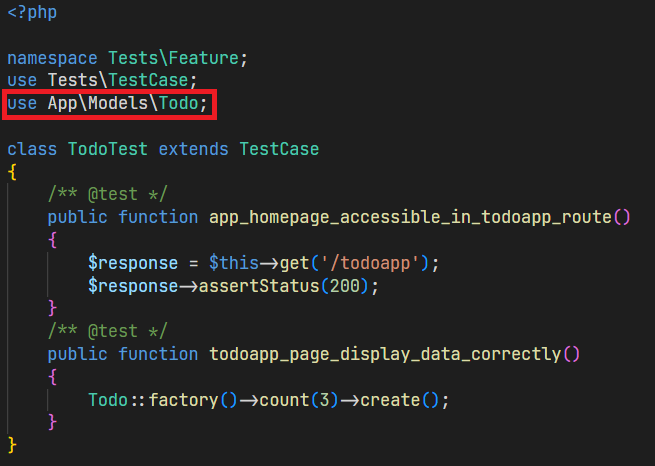
**Assertion** ⇒ là khi Act đã được chạy, Assertion kiểm tra xem Act có hoạt động đúng như mong đợi hay không

**Áp dụng mô hình này vào ứng dụng , ta có quy trình sau**

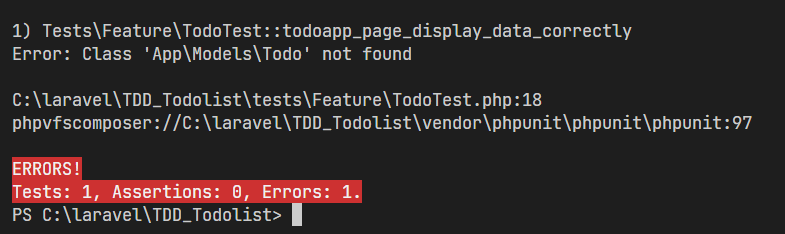
**Arrange** ⇒ seed dữ liệu giả vào database

***Ta add model của Todo vào trong test class , dù model này chưa được khởi tạo***

Ta seed dữ liệu giả vào bằng Factory

[](file:///C:\Users\Tuan\AppData\Local\Temp\Rar$EXa5248.47518\TDD%20trong%20Laravel%20adbeef81a4bf4a308b0e4cfd8c1b2850\Untitled%207.png)

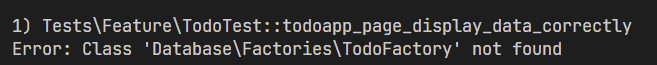
***Ta chạy lại test case***

[](file:///C:\Users\Tuan\AppData\Local\Temp\Rar$EXa5248.47518\TDD%20trong%20Laravel%20adbeef81a4bf4a308b0e4cfd8c1b2850\Untitled%208.png)

***Test case failed vì model Todo không tồn tại, ta tiến hành tạo model Todo bằng lệnh***

***php artisan make:model Todo***

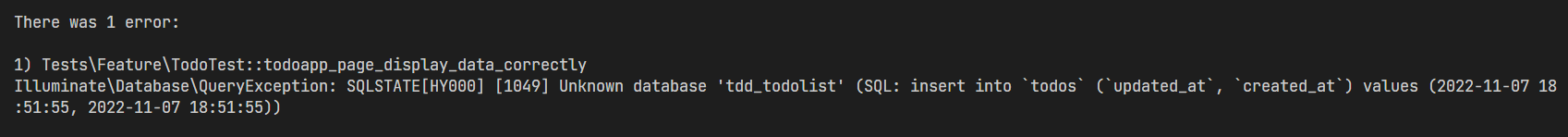
***Ta chạy lại test case***

[](file:///C:\Users\Tuan\AppData\Local\Temp\Rar$EXa5248.47518\TDD%20trong%20Laravel%20adbeef81a4bf4a308b0e4cfd8c1b2850\Untitled%209.png)

***Test case failed vì TodoFactory không tồn tại***

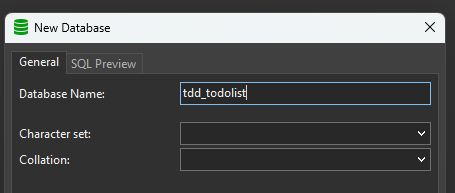
Lại tiếp tục debug bằng cách tạo class Factory mới bằng lệnh

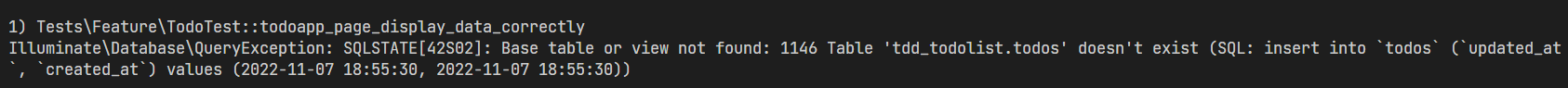
***php artisan make:factory TodoFactory***

[](file:///C:\Users\Tuan\AppData\Local\Temp\Rar$EXa5248.47518\TDD%20trong%20Laravel%20adbeef81a4bf4a308b0e4cfd8c1b2850\Untitled%2010.png)

***Test case failed vì database không tồn tại***

Tạo database bằng phpmyadmin, navicat, hoặc bất cứ trình quản lý database nào bạn có

[](file:///C:\Users\Tuan\AppData\Local\Temp\Rar$EXa5248.47518\TDD%20trong%20Laravel%20adbeef81a4bf4a308b0e4cfd8c1b2850\Untitled%2011.png)

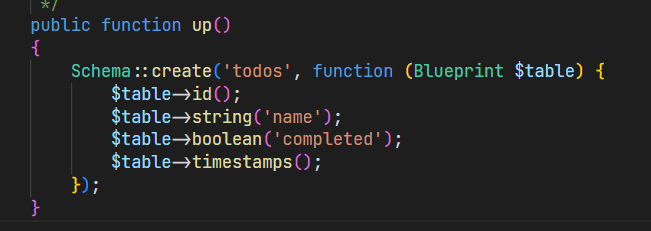
[](file:///C:\Users\Tuan\AppData\Local\Temp\Rar$EXa5248.47518\TDD%20trong%20Laravel%20adbeef81a4bf4a308b0e4cfd8c1b2850\Untitled%2012.png)

***Test case failed vì database không có table cần thiết***

**Tạo table bằng laravel migration**

***php artisan make:migration create\_todos\_table***

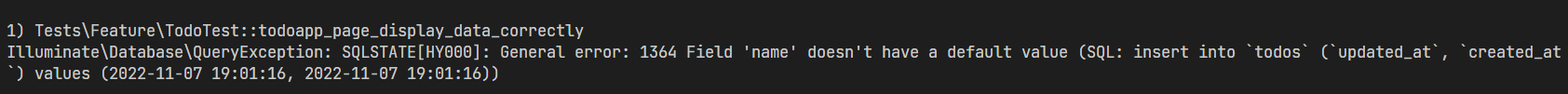
Trong ***database\migrations\{date-time}\_create\_todos\_table.php*** thêm các trường cần thiết cho table

[](file:///C:\Users\Tuan\AppData\Local\Temp\Rar$EXa5248.47518\TDD%20trong%20Laravel%20adbeef81a4bf4a308b0e4cfd8c1b2850\Untitled%2013.png)

**Ta tiến hành Migrate database**

***php artisan migrate***

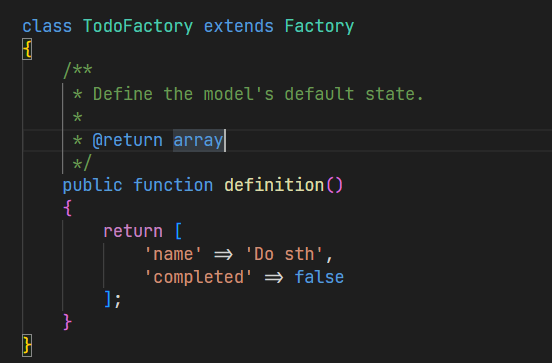
***Lại chạy lại testcase***

[](file:///C:\Users\Tuan\AppData\Local\Temp\Rar$EXa5248.47518\TDD%20trong%20Laravel%20adbeef81a4bf4a308b0e4cfd8c1b2850\Untitled%2014.png)

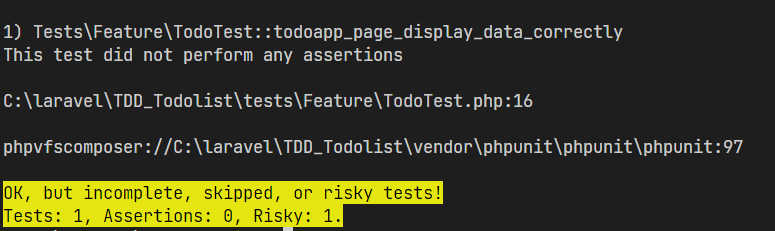
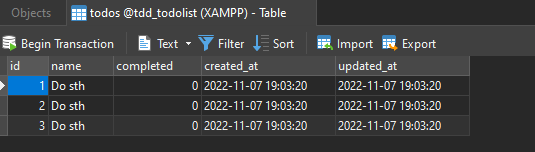
Test case failed vì factory không có structure cụ thể để seed dummy data vào database

Vào file : ***database\factories\TodoFactory.php***

**Them structure definition cho TodoFactory**

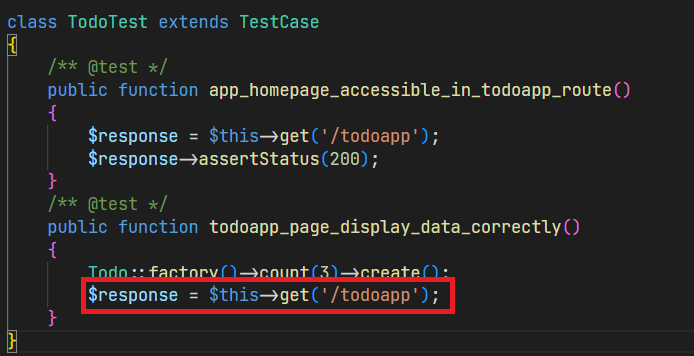
[](file:///C:\Users\Tuan\AppData\Local\Temp\Rar$EXa5248.47518\TDD%20trong%20Laravel%20adbeef81a4bf4a308b0e4cfd8c1b2850\Untitled%2015.png)

***Test case passed***

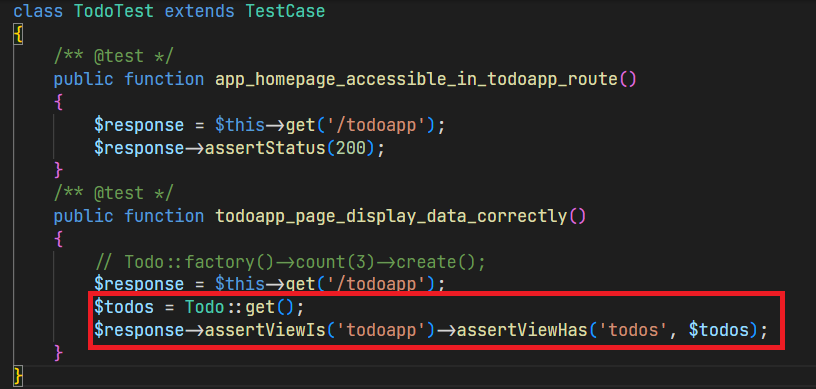
[](file:///C:\Users\Tuan\AppData\Local\Temp\Rar$EXa5248.47518\TDD%20trong%20Laravel%20adbeef81a4bf4a308b0e4cfd8c1b2850\Untitled%2016.png)[](file:///C:\Users\Tuan\AppData\Local\Temp\Rar$EXa5248.47518\TDD%20trong%20Laravel%20adbeef81a4bf4a308b0e4cfd8c1b2850\Untitled%2017.png)

***Database đã được seed dummy data , hoàn thành Arrange parse***

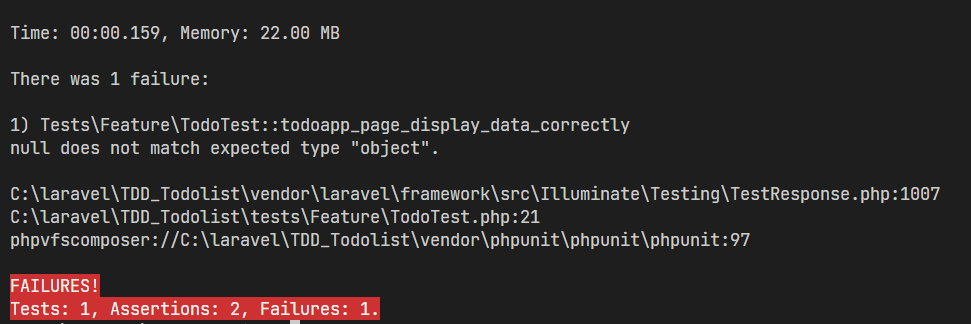
2. **Act** ⇒ redirect về route todoapp

[](file:///C:\Users\Tuan\AppData\Local\Temp\Rar$EXa5248.47518\TDD%20trong%20Laravel%20adbeef81a4bf4a308b0e4cfd8c1b2850\Untitled%2018.png)

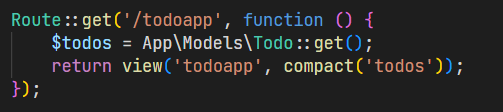
1. **Assertion** ⇒ Kiểm tra xem các todo trong database đã được pass trong view hay chưa

[](file:///C:\Users\Tuan\AppData\Local\Temp\Rar$EXa5248.47518\TDD%20trong%20Laravel%20adbeef81a4bf4a308b0e4cfd8c1b2850\Untitled%2019.png)

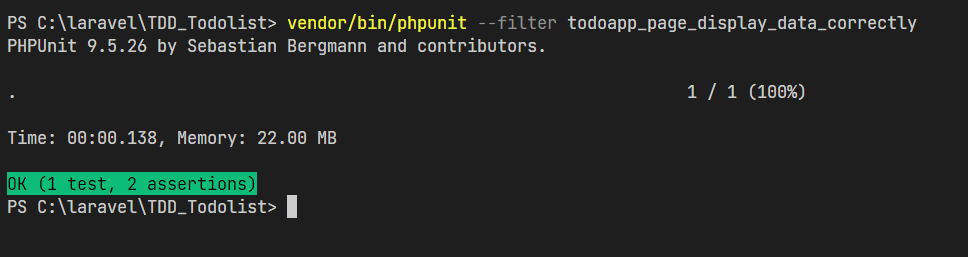
***Test case thông báo failed , không tìm thấy object được pass trong view***

[](file:///C:\Users\Tuan\AppData\Local\Temp\Rar$EXa5248.47518\TDD%20trong%20Laravel%20adbeef81a4bf4a308b0e4cfd8c1b2850\Untitled%2020.png)

***Tiến hành debug web.php , pass object todos sang views***

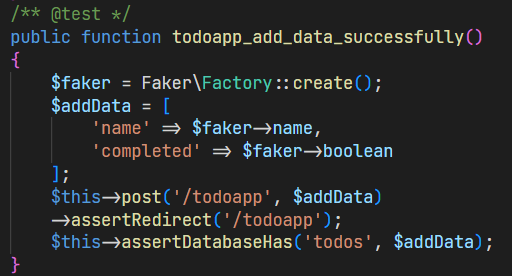
[](file:///C:\Users\Tuan\AppData\Local\Temp\Rar$EXa5248.47518\TDD%20trong%20Laravel%20adbeef81a4bf4a308b0e4cfd8c1b2850\Untitled%2021.png)

***Chạy lại test case , ta thấy test case đã passed***

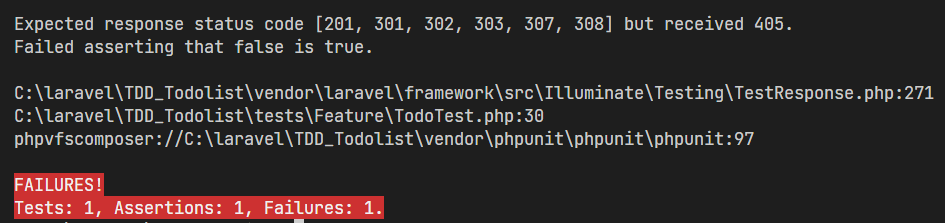
[](file:///C:\Users\Tuan\AppData\Local\Temp\Rar$EXa5248.47518\TDD%20trong%20Laravel%20adbeef81a4bf4a308b0e4cfd8c1b2850\Untitled%2022.png)

**Tiến tục áp dụng TDD cho tất cả các functions và requirements còn lại**

**Test-case: Kiểm tra xem có add dữ liệu được vào database không**

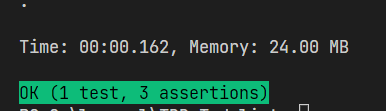


***Lỗi vì route chưa được tạo***

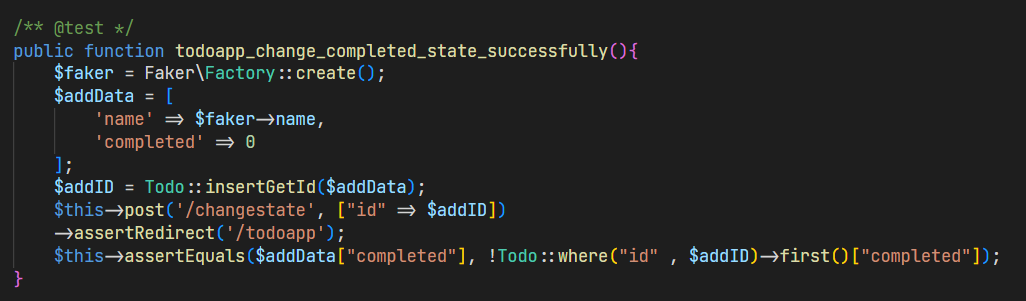


***Tạo route để fix lỗi***

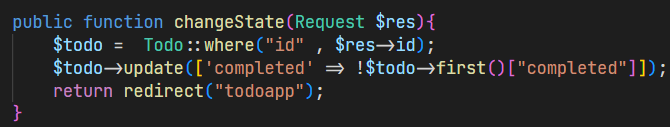




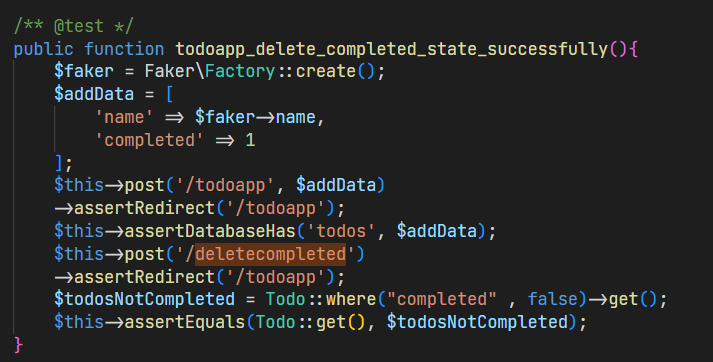
**Test-case: Kiểm tra xem có thay đổi trạng thái completed của todo trong database được không**

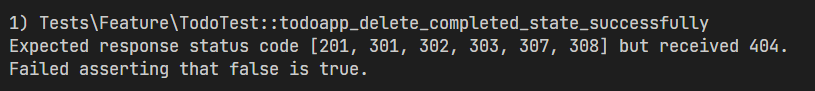




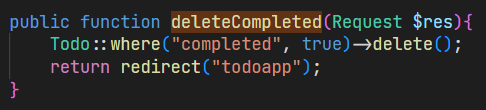


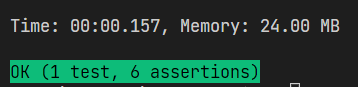
**Test-case: Kiểm tra xem có xóa được tất cả những todos đã hoàn thành hay không**



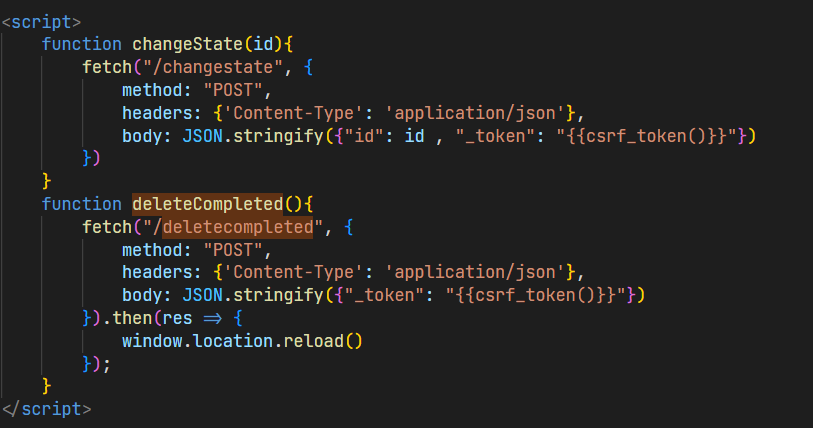






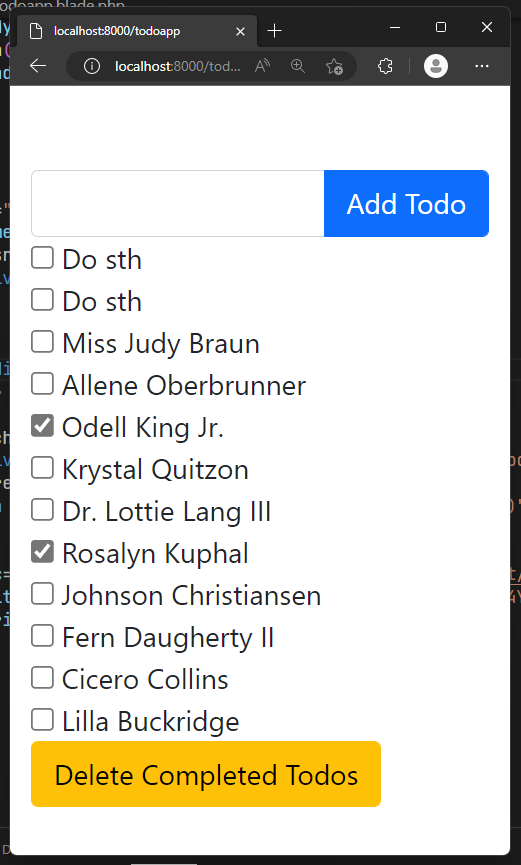


**Merge route, controller đã build theo TDD vào UI**





**Kết quả: App TodoList được xây dựng theo phương pháp TDD**



**Reactor Code**

Ta tìm các cách để rút gọn + cải thiện code ***, ví dụ như*** nhét web route vào controller riêng, ...



