3.9. Covariance & Contravariance in PHP

Khi bạn có một lớp kế thừa từ một lớp khác và ghi đè phương thức, cần có một số kiểm tra kiểu cụ thể để đảm bảo rằng phương thức ghi đè tương thích với phương thức cha. Ví dụ, các kiểu tham số và kiểu trả về của phương thức được ghi đè trong lớp con phải tương thích với phương thức của lớp cha. A screen shot of a computer

Description automatically generated Kiểu dữ liệu biến thiên đảm bảo tính an toàn kiểu dữ liệu để trình biên dịch biết cách xử lý khi gặp phải một lớp con ghi đè phương thức và có kiểu trả về hoặc kiểu tham số khác. Nó cần phải biết nó cần phải phản ứng thế nào, liệu nó đang chuyển từ kiểu trả về ít cụ thể đến kiểu trả về nhiều cụ thể hơn hay ngược lại với kiểu tham số. Ví dụ, PHP cho phép chúng ta ghi đè phương thức để trả về một kiểu cụ thể hơn, được gọi là kiểu dữ liệu trả về tương hợp và nó cũng cho phép chúng ta chấp nhận các kiểu tham số kiểu chung hoặc ít cụ thể hơn, được gọi là kiểu dữ liệu tham số tương phản.

\*Ví dụA screenshot of a computer

Description automatically generated

Đầu tiên, chúng ta có lớp animal ở đây, đó là một lớp trừu tượng cơ bản chỉ có một phương thức trừu tượng duy nhất gọi là speak.



-Sau đó, chúng ta có hai lớp mở rộng từ lớp animal, đó là các lớp chó

A screen shot of a computer

Description automatically generated

và mèo

A screen shot of a computer program

Description automatically generated

và chúng cài đặt phương thức speak theo cách riêng của chúng.

-Chúng ta có một giao diện animal shelter và hai lớp thực hiện giao diện này. A screenshot of a computer program

Description automatically generated

Giao diện có một phương thức duy nhất gọi là adopt, nhận một đối số duy nhất là một chuỗi (tên) và phải trả về một thể hiện của đối tượng animal. A black background with yellow text

Description automatically generated

Bây giờ, lớp cat shelter và dog shelter thực hiện giao diện animal shelter và chúng cài đặt phương thức đó, lớp cat shelter trả về một thể hiện mèo mới A screen shot of a computer

Description automatically generated

và lớp dog shelter trả về một thể hiện hàm dog mới. A screen shot of a computer program

Description automatically generated

-Và lưu ý sự khác biệt giữa các kiểu trả về. Lớp dog shelter, ví dụ, chúng ta có kiểu trả về là chó, trong khi phương thức giao diện có kiểu trả về là animal. Chúng ta thấy rằng kiểu dữ liệu tương hợp cho phép phương thức của con trả về một kiểu cụ thể hơn so với kiểu trả về của phương thức cha và đó chính xác là điều đang xảy ra ở đây. Chúng ta có một phương thức lớp trả về một kiểu cụ thể hơn thay vì kiểu chung của phương thức cha hoặc trong trường hợp này, kiểu chữ ký hợp đồng hoặc giao diện.

Thử nghiệm mở tệp index.phpA computer screen with text

Description automatically generated

và chúng ta đang khởi tạo đối tượng cat shelter A screen shot of a computer

Description automatically generated

và dog shelter ở đây

A screen shot of a computer

Description automatically generated

và chúng ta đang nhận nuôi hai con vật và sau đó chúng ta gọi phương thức speak trên cả hai. Kết quả là con đầu tiên là một con mèo và con thứ hai là một con chó A screen shot of a computer

Description automatically generated vì phương thức adopt từ lớp cat shelter trả về một đối tượng mèo và phương thức adopt từ lớp dog shelter trả về một đối tượng chó.

Vậy để tóm lại, kiểu dữ liệu tương hợp cho phép chúng ta đi từ kiểu ít cụ thể hơn đến kiểu cụ thể hơn trong trường hợp giao diện, chúng ta có kiểu chung là kiểu trả về cho phương thức đó khi chúng ta triển khai giao diện này vào một lớp cụ thể, cho dù đó là cat shelter, dog shelter hoặc bất kỳ loại shelter động vật nào khác, kiểu trả về của phương thức đó có thể là một kiểu cụ thể hơn. Nó có thể là một kiểu tốt hơn thay vì là một kiểu chung. Điều này được cho phép vì chó vẫn là một con cái của lớp động vật và do đó chữ ký phương thức (extend) này tương thích với chữ ký của phương thức cha và trong trường hợp này là chữ ký hợp đồng hoặc giao diện.

- Kiểu dữ liệu tham số tương phản.

\*Ví dụ

- Chúng ta có một vài sửa đổi ở đây cho lớp động vật và lớp chó và chúng ta có hai lớp mới là food và animal food. Lớp food chỉ là một lớp trống A black rectangular object with white lines

Description automatically generated

và sau đó animal food mở rộng từ lớp food A black and orange text

Description automatically generated.

-Lớp food hơi giống như một lớp cơ sở, nó không phải là một lớp trừu tượng, nó chỉ là một lớp cơ sở mà animal food mở rộng từ đó. Sau đó, trong lớp động vật, chúng ta có một phương thức gọi là eat chấp nhận animal food là đối số của nó.

A screen shot of a computer program

Description automatically generated

Trong lớp chó mở rộng từ lớp động vật, chúng ta ghi đè phương thức eat và thay vì chấp nhận một loại animal food cụ thể hơn, nó giờ đây chấp nhận một loại thức ăn tổng quát hơn, một loại ít cụ thể hơn. Vì vậy, nó đang đi từ kiểu cụ thể hơn đến kiểu ít cụ thể hơn trong phương thức ghi đè.

-Chúng ta có một phương thức lớp chấp nhận một loại thức ăn cụ thể hơn thay vì loại thức ăn chung của phương thức cha

A computer screen with text

Description automatically generated

quay lại tệp index.php và thay đổi

A screen shot of a computer program

Description automatically generated

Chúng ta đang tạo một đối tượng animal food và chúng ta đang truyền nó vào phương thức eat của mèo  và chúng ta đang tạo một đối tượng thức eat tổng quát và truyền nó vào phương thức eat của chóA black background with text

Description automatically generated và điều này nên hoạt động.

Chạy mã

A black screen with white text

Description automatically generated

Chạy mã này trong PHP 7.4

A black screen with white text

Description automatically generated

- Để tóm lại, tóm lại, cơ bản, khi chuyển từ ít cụ thể đến cụ thể hơn, đó là kiểu Covariance, và ngược lại, khi chuyển từ cụ thể hơn đến ít cụ thể hơn, đó là kiểu Contravariance trong PHP A blue background with white text

Description automatically generated

# 3.10. Attributes - Simple Router With Attributes

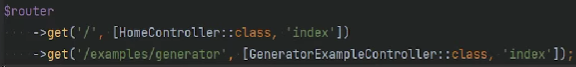
- Thuộc tính (attributes) định nghĩa chính thức cung cấp khả năng thêm thông tin siêu dữ liệu có thể đọc được bằng máy có cấu trúc vào các khai báo trong các lớp mã, các phương thức, hàm, tham số, thuộc tính và hằng số lớp có thể được nhắm mục tiêu của một thuộc tính, siêu dữ liệu được xác định bởi các thuộc tính sau đó có thể được kiểm tra trong thời gian chạy bằng cách sử dụng các thuộc tính apis phản chiếu.

- Ví dụ

A screen shot of a computer program

Description automatically generated

bộ định tuyến rất đơn giản tùy chỉnh hoạt động bằng cách đăng ký các route trong index.php



- Mở phương thức chỉ mục của bộ điều khiển home 

- Thêm siêu dữ liệu thuộc tính vào phương thức này trực tiếp trong bộ điều khiển để cú pháp thuộc tính đặt ký hiệu số rồi đặt dấu ngoặc và bên trong các dấu ngoặc, có thể sử dụng dấu ngoặc đơn và chuyển vào các đối số của hàm tạo

A screenshot of a computer program

Description automatically generated

- Tạo lớp (route) trong không gian tên thuộc tính và bạn có thể đặt nó vào bất kỳ không gian tên nào mong muốn

A screenshot of a computer

Description automatically generated

tạo một hàm tạo và trong mộ hàm tạo, chúng ta cần chấp nhận hai đối số, một là đường dẫn lộ trình thực tế và đường dẫn kia là phương thức yêu cầu cho dù đó là post, get, v.v. và đặt nó ở chế độ mặc định

A screenshot of a computer

Description automatically generated

- Thêm thuộc tính thuộc tính vào lớp, định nghĩa các thuộc tính trên chính lớp đó ngay trước tên lớp

A screenshot of a computer

Description automatically generated

- Quay lại bộ điều khiển và có thể nhập lớp thuộc tính route A screen shot of a computer

Description automatically generated

Có thể thấy rằng lớp phản chiếu có một phương thức gọi là thuộc tính get trả về một mảng đối tượng thuộc tính và sau đó chúng ta có thể gọi tên get và nhận đối số trên đó để có thêm thông tin, thêm một phương thức ở đây gọi là đăng ký các tuyến điều khiển hoặc đăng ký các tuyến từ các thuộc tính hoặc đăng ký các tuyến thuộc tính, v.v.

Gọi phương thức đăng ký các tuyến từ các thuộc tính của bộ điều khiển

A black background with white text

Description automatically generated

Chấp nhận danh sách các bộ điều khiển thông qua hàm tạo trong lớp bộ định tuyến để có thể chấp nhận mảng bộ điều khiển

A black screen with white text

Description automatically generated

Gọi phương thức này từ index.php

A screen shot of a computer program

Description automatically generated

- Gọi phương pháp chuyển vào danh sách các bộ điều khiển để có bộ điều khiển home và sau đó thêm bộ điều khiển generator example controller A screen shot of a computer program

Description automatically generated

quay lại phương pháp này

A black screen with yellow text

Description automatically generated

- Điền thông tin vào để chúng ta cần lặp qua từng bộ điều khiển .Mỗi bộ controller làm bộ điều khiển và đối với mỗi bộ controller, tạo một đối tượng reflectionClass Thực hiện cho mỗi getMethods của reflectionController làm phương thức và sau đó chúng ta có thể sử dụng phương thức get attribute để truy cập các thuộc tính được liên kết với phương thức

A screen shot of a computer program

Description automatically generated

- Chỉ định lớp thuộc tính mà chúng tôi muốn lọc theo vì vậy chúng ta sẽ chuyển vào tên lớp route



- Thực hiện vòng lặp foreach thuộc tính dưới dạng thuộc tính và sau đó có thể sử dụng phương thức phiên bản mới để thực sự khởi tạo lớp thuộc tính route bằng attributes -> newInstance

A screen shot of a computer

Description automatically generated

- Thực hiện register và chúng ta cần chuyển vào method và sau đó chúng ta cần chính route đó nên chúng ta sẽ thực hiện một route dẫn routepath chuyển bộ điều khiển và method vì vậy chúng ta sẽ thực hiện bộ điều khiển và tên phương thức bằng cách gọi phương thức getname trên đối tượng phương thức



- Kết quả

A close up of a text

Description automatically generated

- Kết nối mảng route để thể thấy các route được đăng ký ở đó mà không cần phải truy cập từng route mà chúng tôi sẽ đăng ký thông qua bộ điều khiển, thêm một số thẻ trước



- Làm mới trang và có thể thấy đây là route duy nhất hiện đã được đăng ký

A close up of text

Description automatically generated

- Có thể thêm route và chuyển vào ‘/example/generator’ và kiểm tra, có thể thấy route đó đã được đăng ký A white background with black text

Description automatically generated

- Thêm một vài route nữa thêm các route post  và đặt vì vậy chúng ta sẽ đóng cái này lại, chúng ta sẽ quay lại bộ home controler, hãy thêm một số phương thức lưu trữ và hiện tại nó sẽ không làm gì cả, chúng ta sẽ chỉ thêm route ở đây và chúng ta sẽ chuyển bài đăng ở đây và điều này không làm được gì Ở đây không cần phải mã hóa cứng, chúng ta có thể sử dụng các enum đã được giới thiệu trong php 8.1 nhưng đó không phải là vấn đề nên bây giờ chúng ta sẽ giữ nguyên nó, hãy thêm update  và chúng ta sẽ thực hiện việc này, hãy làm mới trang này và như bạn có thể thấy, chúng tôi có post route bài và chúng tôi có put routeA screenshot of a computer program

Description automatically generated nên hãy tiếp tục, bạn có thể sử dụng tính kế thừa và triển khai các giao diện trong thuộc tính của mình nếu cần, ví dụ: chúng tôi có thể tạo giao diện route và triển khai giao diện đó trong thuộc tính route; chúng tôi sẽ mở lớp route ở đây và chúng ta có thể triển khai một số loại giao diện route  , sau đó chúng ta có thể tạo giao diện này và đặt nó trong thư mục contract hoặc interfaceA screenshot of a computer

Description automatically generated

và mọi thứ vẫn hoạt động như mong đợi, chúng ta cũng có thể tạo các lớp con cho cụ thể yêu cầu các phương thức nên hiện tại chúng tôi đang mã hóa post và đặt ở đây  nhưng nếu chúng tôi có thể làm điều gì đó như post và put và nhận để chúng tôi có thể loại bỏ điều này về cơ bản và chúng tôi muốn thực hiện công việc này theo cách chúng tôi sẽ thực hiện công việc này thì chúng tôi có thể tạo tất cả ba lớp thuộc tính này và chúng tôi có thể mở rộng lớp route vì vậy chúng tôi sẽ tạo lớp này và thêm nó vào trong không gian tên thuộc tính A screenshot of a computer

Description automatically generated

và chúng tôi sẽ mở rộng thuộc tính route và chúng tôi cũng sẽ biến lớp này thành một thuộc tính vì vậy chúng ta hãy làm tương tự với post và put và tạo các lớp này sau đó trong mỗi lớp chúng ta cần có hàm tạo đúng và trong hàm tạo, chúng ta không cần chấp nhận phương thức vì phương thức này sẽ được truyền cho cấu trúc cha mẹ nên chúng ta sẽ gọi hàm tạo cha mẹ ở đây và chuyển đường dẫn route và chúng ta sẽ chuyển put làm đối số thứ hai sau đó chúng ta sẽ sao chép cái này

A computer screen shot of a code

Description automatically generated

và đặt nó vào post và get và chúng ta sẽ thay đổi nó thành post

A screen shot of a computer program

Description automatically generated

và chúng ta sẽ thay đổi phần này thành get A screen shot of a computer program

Description automatically generated ngay bây giờ nếu chúng ta quay lại bộ home controler, chúng ta sẽ có thể nhập những thứ này và mọi thứ vẫn hoạt động và tất nhiên việc sử dụng thuộc tính route cũng sẽ hoạt động và chúng ta đang sử dụng nó trên bộ điều khiển generator nên chúng ta sẽ thấy rằng nó thực sự hoạt động, hãy làm mới trang ở đây và có vẻ như nó không hoạt động. route được đăng ký duy nhất là route mà chúng tôi sử dụng trực tiếp lớp route và đó là lớp trình tạo, nó chưa đăng ký bất kỳ tuyến nào trong số này và tại sao lại như vậy Nó không hoạt động, lý do nó không hoạt động là vì chúng ta đang yêu cầu php trong lớp bộ định tuyến của chúng ta ở đây lấy các thuộc tính của lớp route nhưng các thuộc tính trong bộ home controler không chính xác là các đối tượng route mặc dù chúng mở rộng lớp thuộc tính route. điều chúng ta cần làm là chúng ta cần yêu cầu phương thức phản chiếu cung cấp cho chúng ta bất kỳ thuộc tính nào mở rộng lớp bộ route hoặc bất kỳ thuộc tính nào là thể hiện của lớp route. Nếu chúng ta mở lại phương thức thuộc tính get, chúng ta thấy rằng nó có đối số tùy chọn thứ hai ở đây được gọi là flags và đây là tiêu chí mà thuộc tính được tìm kiếm, điều đó có nghĩa là chúng ta thực sự có thể truyền một hằng số cờ để chỉ định rằng chúng ta muốn bất kỳ thuộc tính nào là thể hiện của lớp route và hằng số đó là thuộc tính phản chiếu là thể hiện của bất kỳ thuộc tính nào. Lớp thuộc tính là phiên bản của lớp route sẽ được trả về nếu chúng ta quay lại trình duyệt và làm mới trang, chúng ta thấy rằng mọi thứ đã trở lại bình thường và tất cả các route này đã được đăng ký lại, được rồi, còn một vài điều nữa mà tôi muốn Điều cần đề cập trước khi chúng ta kết thúc phần này là chúng ta có thể giới hạn hoặc hạn chế nơi một thuộc tính có thể được sử dụng, ví dụ: chúng ta có thể nói rằng thuộc tính get này  ở đây chỉ có thể được sử dụng trên các phương thức vì chúng ta đang sử dụng chúng trên các phương thức ngay bây giờ và ở đó không có lý do gì để sử dụng thuộc tính này ở bất kỳ nơi nào khác ngoài các phương thức nếu chúng ta vào lớp get ở đây, chúng ta thấy rằng chúng ta đang sử dụng thuộc tính đó  trên lớp, điều đó có nghĩa là chúng ta có thể sử dụng các thuộc tính trên lớp và chúng ta có thể Để sử dụng các thuộc tính trên các phương thức, bây giờ có thêm một vài nơi có thể sử dụng thuộc tính. Hãy mở chính lớp thuộc tính ở đây A screen shot of a computer

Description automatically generated và xem những gì chúng ta có và trong hàm tạo ở đây , chúng ta thấy rằng nó chấp nhận một đối số tùy chọn được gọi là flag và theo mặc định, nó được đặt thành nhắm mục tiêu tất cả vì vậy đây là những hạn chế mà chúng ta có thể đặt ra, chúng ta có thể nói với thuộc tính của mình rằng nó chỉ có thể được sử dụng trên các lớp, nó chỉ có thể được sử dụng khi được sử dụng trên các hàm, nó chỉ có thể được sử dụng trên các phương thức trên các thuộc tính trên các hằng số lớp trên các tham số hoặc về mọi thứ và chúng ta sẽ nói về tính uy tín trong giây lát vì vậy điều chúng ta có thể làm là chúng ta có thể tùy ý chuyển vào đây và hạn chế nơi có thể sử dụng thuộc tính này để chúng ta có thể nói rằng thuộc tính này chỉ có thể được sử dụng trên các phương thức  nên chúng ta Tôi sẽ thực hiện phương thức đích thuộc tính và bây giờ nếu thuộc tính get được sử dụng ở bất kỳ nơi nào khác ngoài phương thức thì nó sẽ đưa ra một ngoại lệ nhưng điều đáng chú ý là nó sẽ chỉ ném ngoại lệ khi phiên bản mới được tạo nên hãy kiểm tra điều này. sẽ thêm phương thức get vào một thứ không phải là phương thức nên chúng ta sẽ đi vào bộ điều khiển chính ở đây và giả sử chúng ta thêm một số loại thuộc tính vào đây  và giả sử rằng vì bất kỳ lý do gì chúng ta muốn thêm thuộc tính get vào thuộc tính này như bạn có thể thấy ide đã gạch chân phần này và nếu chúng ta di chuột qua nó, nó sẽ thông báo rằng thuộc tính đó không thể được áp dụng cho một thuộc tính vì bây giờ nó không chứa thuộc tính mục tiêu flag A screenshot of a computer

Description automatically generated

nếu chúng tôi làm mới trang, bạn có nghĩ rằng chúng tôi sẽ làm như vậy không nhận được một ngoại lệ, chúng tôi làm mới trang và không, chúng tôi không nhận được bất kỳ ngoại lệ nào, lý do là nếu bạn nhớ vài phút trước tôi đã nói rằng việc xác thực chỉ xảy ra khi phiên bản mới được tạo, điều đó có nghĩa là trong lớp route ở đây  khi phiên bản mới này được gọi là khi quá trình xác thực thực tế xảy ra và điều này thực sự không bao giờ được gọi cho thuộc tính này bởi vì chúng tôi đang lặp lại các phương thức của bộ điều khiển mà chúng tôi không nhận được các thuộc tính và do đó chúng tôi không khởi tạo đối tượng đó ở đây và do đó nó không phải vậy thực hiện xác thực hợp lệ và nó không đưa ra bất kỳ ngoại lệ nào để kích hoạt ngoại lệ đó. Điều chúng ta cần làm là gọi phiên bản mới trên thuộc tính thuộc tính. Chúng ta có thể kiểm tra nhanh điều đó ở đây, chúng ta có thể lấy các thuộc tính của lớp phản chiếu để chúng ta có thể lấy các thuộc tính hoặc chúng ta có thể lấy một thuộc tính cụ thể và trong trường hợp này, chúng ta có thể lấy thuộc tính cụ thể vì chúng ta biết đó là thuộc tính x và chúng ta sẽ đặt thuộc tính đó thành thuộc tính  và sau đó chúng ta cần lấy thuộc tính của thuộc tính nên chúng tôi sẽ lấy các thuộc tính và chúng tôi sẽ thực hiện điều tương tự như thế này nhưng chúng tôi sẽ thay thế phương thức bằng thuộc tính và sau đó chúng tôi sẽ thực hiện vòng lặp tương tự ở đây A black screen with text

Description automatically generated

và lưu ý rằng chúng tôi thực sự sẽ loại bỏ thuộc tính đó Tôi chỉ làm điều này để chứng minh ngoại lệ sẽ được ném ra và chúng tôi không cần điều này vì chúng tôi không đăng ký bất kỳ route nào ở đây, chúng tôi sẽ tạo một phiên bản mới và route kết xuất var để chúng tôi có thể xem liệu nó thực sự hoạt động, hãy làm mới trang và bây giờ đây là lúc chúng tôi gặp phải lỗi nghiêm trọng như bạn có thể thấy nó nói rằng thuộc tính get không thể nhắm mục tiêu thuộc tính được phép chỉ là phương thức A close up of a text

Description automatically generated nếu chúng tôi xóa phần này

A blue screen with yellow text

Description automatically generated và hoàn nguyên nó về vị trí ban đầu làm mới trang mà chúng tôi không gặp bất kỳ lỗi nào, đó là lý do tại sao tôi đã nói trước đó rằng việc xác thực chỉ xảy ra khi phiên bản mới được tạo và đây là một ví dụ về điều đó nên chúng tôi sẽ loại bỏ việc các hằng số này thực chất là mặt nạ bit A screen shot of a computer screen

Description automatically generated, nghĩa là chúng tôi có thể sử dụng các thao tác bitwise trên chúng để chúng tôi có thể nói rằng chúng tôi muốn thuộc tính cụ thể chỉ được nhắm mục tiêu ở lớp và phương thức hoặc chúng tôi có thể nói rằng nó chỉ có thể được sử dụng trên các thuộc tính và hằng số lớp, v.v. có thể nói rằng nó có thể được sử dụng trong phương thức hoặc nó có thể được sử dụng trong lớp và điều này sẽ hiệu quả  nếu bạn cần ôn lại các toán tử bitwise, hãy xem bài học trong loạt bài đầu tiên có liên kết trong phần mô tả nhé. Hãy hoàn nguyên điều này và chuyển sang hằng số có thể lặp lại ở đây theo mặc định, chúng ta chỉ được phép sử dụng thuộc tính một lần cho mỗi khai báo, hãy để tôi chỉ cho bạn ý của tôi vì vậy hãy nói rằng chúng tôi muốn có một route khác lắng nghe phương thức này, chẳng hạn như vậy chúng tôi đang nói rằng khi ai đó truy cập trang chủ của chúng tôi, đây chỉ là một đường gạch chéo (/), chúng tôi thực hiện phương pháp này nếu chúng tôi muốn làm điều tương tự khi ai đó truy cập trang chủ như bạn có thể thấy ide ở đây nhấn mạnh điều này  và cho chúng tôi biết thuộc tính đó là không thể lặp lại vì nó không chứa cờ có thể lặp lại nếu chúng ta làm mới trang, chúng ta sẽ gặp lỗi này. Để giải quyết vấn đề này, chúng ta cần sử dụng cờ có thể lặp lại để nếu quay lại thuộc tính get ở đây, chúng ta có thể sử dụng bitwise hoặc Thuộc tính toán tử và pass bây giờ có thể lặp lại nếu chúng ta quay lại bộ điều khiển chính, chúng ta thấy phần gạch chân đã biến mất và nếu chúng ta làm mới trang thì mọi thứ sẽ trở lại bình thường và chúng ta thấy rằng cả dấu gạch chéo và dấu gạch chéo home đều trỏ đến phương thức chỉ mục trên home bộ điều khiểnA close-up of a computer screen

Description automatically generated xin lưu ý rằng cả đối số vị trí và đối số được đặt tên đều có thể được sử dụng khi sử dụng thuộc tính và truyền đối số ở đây ngay bây giờ. Chúng tôi đang sử dụng đối số vị trí nhưng chúng tôi có thể sử dụng đối số được đặt tên nếu chúng tôi muốn, đặc biệt là khi bạn có nhiều đối số mà bạn đang chuyển đến đây. đôi khi nên sử dụng các đối số được đặt tên để ở đây chúng ta có thể nói điều gì đó như routepath  và điều này sẽ vẫn hoạt động nếu chúng ta làm mới trang, chúng ta thấy rằng mọi thứ đều hoạt động. Ngoài ra, xin lưu ý rằng bạn có thể có nhiều thuộc tính khác nhau cho mỗi khai báo mà bạn không cần có thể lặp lại để được bật vì điều đó có thể lặp lại áp dụng cho thuộc tính cụ thể khi bạn đang sử dụng nó nhiều lần, ví dụ như thuộc tính get, chúng tôi không thể sử dụng nó nhiều lần nếu không bật tùy chọn lặp lại trên lớp get, tuy nhiên bạn có thể sử dụng được kết hợp với bất kỳ thuộc tính nào khác để chúng tôi có thể nhận và sau đó thực hiện điều gì đó như route và thực hiện thêm head hoặc bạn có thể để là post  và điều này cũng hiệu quả vì chúng tôi không lặp lại thuộc tính cụ thể mà chúng tôi đang sử dụng các thuộc tính khác nhau nên nếu tôi loại bỏ Thuộc tính get này ở đây và chúng tôi xóa tùy chọn có thể lặp lại ở đây  và chúng tôi làm mới trang, chúng tôi thấy rằng mọi thứ đang hoạt động như mong đợi và chúng tôi không gặp phải bất kỳ lỗi nghiêm trọng nào. Chúng tôi thấy rằng yêu cầu head đã được đăng ký cũng như yêu cầu post cho route về nhà vì vậy chúng tôi sẽ thêm nó trở lại và chúng tôi sẽ sao chép cái này và dán nó vào các thuộc tính khác vì vậy chúng tôi sẽ đặt nó vào vị trí bài đăng và trong route cũng như một điều khác mà tôi muốn cho phép bạn biết rằng có một vài thuộc tính tích hợp trong php và tôi chắc chắn rằng sẽ có nhiều thuộc tính hơn đi kèm với các phiên bản mới hơn. Các id như phpstorm cũng cung cấp các thuộc tính riêng của chúng nhưng hai thuộc tính mới là #[Deprecated] và #[Jit] nên điều này Đây có phải là bài học này không tôi hy vọng bạn thích nó nếu bạn thích, vui lòng cho video này một lượt thích và đăng ký nếu bạn chưa đăng ký, vì vậy hãy nhấn vào biểu tượng chuông thông báo để đảm bảo rằng bạn không bỏ lỡ bất kỳ video nào trong tương lai của tôi cảm ơn bạn rất nhiều vì đã xem và ủng hộ tất cả và tôi sẽ gặp bạn ở phần tiếp theo.