



TRUNG TÂM HUẤN LUYỆN LẬP TRÌNH

Khai giảng các khóa tháng 4: Lập trình **RUBY** - Lập trình **PHP** - Lập trình **ANDROID**

lxc @lxc

Follow

★ 136 👤 3 📝 6

Published Nov 20th, 2018 8:44 AM

👁 771 💬 0 🔖 3

TABLE OF CONTENTS

I. POST với GET khác nhau như thế nào?

1. phương thức GET là gì?
2. Phương thức POST là gì?
3. Giống nhau.
4. So sánh phương thức GET và POST.

II. Kết luận.

SUGGESTED ORGANIZATIONS



PHP Education Team

📝 12 👤 38 📑 94



+3



So sánh phương thức get và post

post

get;

so sánh get và post

get là gì?

post là gì?

...

I. POST với GET khác nhau như thế nào?

Trong lập trình web. Để xử lý việc nhận gửi thông tin từ 1 form của người dùng nhập vào là việc rất thường xuyên. Chúng ta thường sử dụng 2 phương thức POST và GET. Tuy nhiên lúc nào sử dụng POST. lúc nào sử dụng GET? Sau đây là sự giống nhau và



+3



1. phương thức GET là gì?

- Phương thức GET rất dễ nhận thấy đó là trên URL sẽ kèm theo dữ liệu mà chúng ta muốn gửi.
- Client gửi lên Phương thức GET là phương thức gửi dữ liệu thông qua đường dẫn URL nằm trên thanh địa chỉ của Browser. Server sẽ nhận đường dẫn đó và phân tích trả về kết quả cho bạn. Server sẽ phân tích tất cả những thông tin đằng sau dấu hỏi (?) chính là phần dữ liệu mà Client gửi lên.
- Ví dụ: Với URL viblo.asia?id=10 thì Server sẽ nhận được giá trị id = 10
- Để truyền nhiều dữ liệu lên Server ta dùng dấu & để phân cách giữa các cặp giá trị. Giả sử tôi muốn truyền id = 10 và title = 'get' thì URL sẽ có dạng viblo.asia?id=10&title=get

`id=10&title=get`. Lưu ý với các bạn là vị trí các cặp giá trị không quan trọng, nghĩa là cặp title có thể nằm trước cặp id cũng được. Server nhận dữ liệu Tất cả các dữ liệu mà Client gửi lên bằng phương thức GET đều được lưu trong một biến toàn cục mà PHP tự tạo ra đó là biến `$_GET`, biến này là kiểu mảng kết hợp lưu trữ danh sách dữ liệu từ client gửi lên theo quy luật key => value. Ví dụ với URL viblo.asia?id=10&title=get

id=10&title=get thì biến \$_GET sẽ có giá trị như sau:

```
$_GET = array(
    'id' => '10',
    'title' => 'get'
);
```

- Lưu ý: Trước khi lấy một dữ liệu nào đó bạn phải kiểm tra tồn tại không không mới lấy nhé, vì nếu bạn không kiểm tra thì giả sử người dùng không truyền dữ liệu qua mà bạn lại nhận thì sẽ bị báo lỗi ngay. Để kiểm tra ta dùng hàm `isset($tenbien)` trong php.

Ví dụ:

```
if (isset($_GET['id'])){
    $id = $_GET['id'];
}
```

2. Phương thức POST là gì?

- Phương thức POST có tính bảo mật hơn vì dữ liệu gửi phải thông qua một form HTML nên nó bị ẩn, nghĩa là chúng ta không thể thấy các giá trị đó được.
- Client Gửi Lên với phương thức GET thì dữ liệu được thấy trên URL thì phương thức POST thì hoàn toàn ngược lại. POST sẽ gửi dữ liệu qua một cái form HTML và

checkbox, password, textarea, number, và được nhận dạng thông qua tên (name) của các input đó. Server nhận dữ liệu Tất cả các dữ liệu gửi bằng phương thức POST đều được lưu trong một biến toàn cục `$_POST` do PHP tự tạo ra, vì thế để lấy dữ liệu thì bạn chỉ cần lấy trong biến này là được. Cũng như lưu ý với các bạn là trước khi lấy phải dùng hàm `isset($bien)` để kiểm tra có hay không nhé.

```
if (isset($_POST['id'])){
    $id = $_POST['id'];
}
```

3. Giống nhau.

- GET và POST đều là hai phương thức của giao thức HTTP.
- Điều gửi dữ liệu về server xử lí, sau khi người dùng nhập thông tin vào form và thực hiện submit.
- Trước khi gửi thông tin, nó sẽ được mã hóa bằng cách sử dụng một giản đồ gọi là url encoding. Giản đồ này là các cặp name/value được kết hợp với các kí hiệu = và các kí hiệu khác nhau được ngăn cách bởi dấu &. Các khoảng trống được xóa bỏ, thay thế bằng kí tự + và bất kì kí tự không phải dạng số và chữ được thay thế bằng giá trị hexa. Sau khi thông tin được mã hóa, nó sẽ được gửi lên Server.

GET	POST
Phương thức GET gửi thông tin người dùng đã được mã hóa được phụ thêm vào yêu cầu trang, truyền thông tin thông qua url.	Phương thức POST truyền thông tin thông qua HTTP header
Dữ liệu của METHOD GET gửi đi thì hiện trên thanh địa chỉ (URL) của trình duyệt.	Dữ liệu được gửi đi với METHOD POST thì không hiển thị trên thanh URL
HTTP GET có thể được cache bởi trình duyệt	HTTP POST không cache bởi trình duyệt
HTTP GET có thể duy trì bởi lịch sử đó cũng là lý do mà người dùng có thể bookmark được.	HTTP POST không thể duy trì bởi lịch sử đó cũng là lý do mà người dùng không thể bookmark HTTP POST được.
Không bảo mật	Bảo mật

cầu đó webbrowser sẽ xem trong cached có kết quả tương ứng với yêu cầu đó không và trả về ngay không cần phải thực thi các yêu cầu đó ở phía server.	luôn thực thi và trả về kết quả cho client
Phương thức GET được giới hạn gửi tối đa chỉ 2048 ký tự	Phương thức POST không có bất kì hạn chế nào về kích thước dữ liệu sẽ gửi.
Không gửi được nhị phân.	Phương thức POST có thể sử dụng để gửi ASCII cũng như dữ liệu nhị phân.
Không bao giờ sử dụng phương thức GET nếu gửi password hoặc thông tin nhạy cảm lên Server.	Dữ liệu gửi bởi phương thức POST thông qua HTTP header, vì vậy việc bảo mật phụ thuộc vào giao thức HTTP. Bằng việc sử dụng Secure HTTP, bạn có thể chắc chắn rằng thông tin của mình là an toàn.

VIBLO

Posts Questions Discussions

Search Viblo



Sign In/Sign up

truy cập tất cả các thông tin đã được gửi bởi phương thức GET.	truy cập tất cả các thông tin được gửi bằng phương thức POST.
Dữ liệu gửi bởi phương thức GET có thể được truy cập bằng cách sử dụng biến môi trường QUERYSTRING.	Không thể
Gửi lại form Với form gửi đi bằng phương thức GET bạn có thể gửi lại bằng cách bấm phím F5 hoặc Ctrl + R	nếu bạn muốn thực hiện việc gửi lại dữ liệu của form thì trình duyệt sẽ hiển thị một hộp thoại cảnh báo. Trở lại trang trước
Dữ liệu gửi đi được lưu lại trong lịch sử web và có thể xem lại	Không được lưu lại trong lịch sử
Trong trường hợp bạn đã gửi form dữ liệu đi rồi sau đó bấm phím Backspace để quay lại trang trước thì với phương thức GET bạn sẽ vẫn được cùng một nội dung (chứa form).	với POST thì bạn sẽ thấy một trang trống.



+3



phương thức GET để truy xuất và xử lý nhanh hơn.

thì thường sử dụng phương thức POST

dữ liệu không cần bảo mật thì dùng phương thức GET

dữ liệu bảo mật thì dùng phương thức POST.



II. Kết luận.

Vậy là qua bài so sánh giữa GET và POST bạn đã hình dung được trường hợp nào nên dùng GET trường hợp nào nên dùng POST chưa. Bài viết còn nhiều thiếu sót mong mọi người góp ý để hoàn thiện hơn. Mọi thắc mắc mọi người comment bên dưới nhé. thanks!

- link tham khảo: <https://freetuts.net/phuong-thuc-get-va-post-trong-php-19.html>



Related

Tuyến

1070 4 1 0

stackjava

2222 3 0 1

Hoàng Ngọc Đại

1254 2 0 0

Hòa Phú

1614 1 0 3

More from lxc

lxc

97 1 0 1

lxc

401 5 5 4

lxc

380 2 0 2

lxc

1496 1 2 2

Comments





Login to comment

RESOURCES

Posts
Questions
Videos
Discussions
Tools

Organizations
Tags
Authors
Recommend System
Machine Learning

LINKS

 Facebook
 GitHub
 Browser extension
 Atom plugin

MOBILE APP



WELCOME TO VIBLO!

Let [register a Viblo Account](#) to get more interesting Posts from [lxc!](#)

Join Us

