

Hướng dẫn sử dụng WP_Query lấy bài viết ngẫu nhiên

Bước 1: Xem thông tin tất cả bài post

Kết quả của query sẽ được lưu vào đối tượng **\$wp_query**.

Kiểm tra đối tượng **\$wp_query** bằng cách chèn đoạn code dưới đây vào file template **index.php** trong theme:

```
<?php

echo "<pre>";

print_r($wp_query);

echo "</pre>";

?>
```

Thông tin bài post nằm trong index post như hình bên dưới

```
[queried_object] =>
[queried_object_id] =>
[request] => SELECT SQL_CALC_FOUND_ROWS wp_posts.ID FROM
[posts] => Array
(
    [0] => WP_Post Object
        (
            [ID] => 29
            [post_author] => 1
            [post_date] => 2021-12-30 01:54:45
            [post_date_gmt] => 2021-12-30 01:54:45
            [post_content] =>
```

"hello world"

Để xem dữ liệu của posts ta trở tới thuộc tính **posts** như ví dụ bên dưới:

```
print_r ($wp_query->posts);
```

hoặc sử dụng biến **\$posts** như sau:

```
print_r ($posts);
```

Để lấy ra chi tiết một bài post, ta dùng biến **\$post**

Bước 3: Lấy ra thông tin tất cả bài post

Để hiển thị toàn bộ bài viết, ta phải dùng vòng lặp while như sau:

```
if( $wp_query->have_posts() ) {  
  
    while( $wp_query->have_posts() ) {  
  
        $wp_query->the_post();  
  
        echo $post->post_title . '<br>';  
  
    }  
  
}
```

Trong đó:

- **have_posts()** là một phương thức của class WP_Query, dùng kiểm tra xem còn bài viết nào trong query hay không.
- **the_post()** chỉ mục để gọi post kế tiếp trong vòng lặp, nếu không khai báo phương thức này thì vòng lặp sẽ bị lặp không có điểm dừng.
- **post_title** là tiêu đề bài viết

Ngoài ra, ta có thể viết ngắn gọn như sau:

```
<?php  
if (have_posts()): while (have_posts()): the_post();  
    echo $post->post_title . '<br>';  
    endwhile;endif;  
  
?>
```

Bước 4: Sử dụng template tags để lấy thông tin bài post

Ở bước 3, để lấy ra tiêu đề bài post, ta dùng `$post->post_title`

Ngoài ra, ta có thể sử dụng template tags **the_title** để lấy thông tin bài viết nhanh hơn như sau:

```
<?php  
if (have_posts()): while (have_posts()): the_post();  
  
    echo the_title('<br>');  
  
    endwhile;endif;
```

```
?>
```

Xem thêm các template tags: https://codex.wordpress.org/Template_Tags

Bước 5: Tùy chỉnh truy vấn để lấy ra các bài post ngẫu nhiên

Để tùy chỉnh câu truy vấn, ta tạo một đối tượng query mới từ class WP_Query như sau:

```
$args = array(
    'post_type'=>'post', //Loại là post
    'orderby' => 'rand', //Lấy ngẫu nhiên
    'posts_per_page' => 5 //Lấy 5 bài post 1 trang
);
$query = new WP_Query( $args );
```

Tham khảo thêm các tham số wp_query tại:

https://developer.wordpress.org/reference/classes/wp_query/

Code hoàn chỉnh:

```
<?php
$args = array(
    'post_type'=>'post', //Loại là post
    'orderby' => 'rand', //Lấy ngẫu nhiên
    'posts_per_page' => 5 //Lấy 5 bài post 1 trang
);
$query = new WP_Query( $args );
if ($query->have_posts()):
    while ($query->have_posts()):
        $query->the_post();
        echo the_title("<br>");
    endwhile;endif;
?>
```

Tài liệu tham khảo

1. https://developer.wordpress.org/reference/functions/add_action/
2. https://developer.wordpress.org/reference/functions/add_filter/
3. https://codex.wordpress.org/Template_Tags
4. https://developer.wordpress.org/reference/classes/wp_query/