**PHP BUỔI 3: PHƯƠNG THỨC TRUYỀN DỮ LIỆU GET, JSON,**

**DỰ ÁN WEBSITE ĐA GIAO DIỆN – ĐA NGÔN NGỮ (VIỆT-ANH)**

I.Phương thức truyền dữ liệu GET:

1. **Phương thức GET**.

* **Lý Thuyết**

*Phương thức GET gửi dữ liệu từ client lên server thông qua các paramter( tham số) trên URL, từ đó server sẽ phân tích dữ liệu để thực thi một hành động nào đó. Các tham số mà phương thức GET gửi lên sẽ bắt đầu sau dấu ? . Và sẽ nối nhau giữa các khóa thông qua dấu &*

VD1: Với url:

|  |
| --- |
| http://vietpro.com?id=18 |

Lúc này chúng ta đang gửi dữ liệu bằng phương thức GET lên server với tham số id=18.

VD2: Với url:

|  |
| --- |
| http://vietpro.com?age=25&name=LeThanhTuan |

Lúc này chúng ta đang gửi dữ liệu bằng phương thức GET lên server với tham số age=25 và name=LeThanhTuan

* **SERVER nhận dữ liệu :**

Để thông báo cho server nhận dữ liệu GET khi có thì chúng ta chỉ cần sử dụng **$\_GET** đây là một biến toàn cục lưu trữ dưới dạng mảng bất tuần tự.

VD3: như với URL ở VD2 thì trên server sẽ lưu trữ cho chúng ta như sau:

|  |
| --- |
| $\_GET = [  'age' => 25,  'name' => 'LeThanhTuan'  ];  //Như thế thì chúng ta chỉ cần lấy giá trị theo kiểu mảng là xong  echo $\_GET['age']; // lấy age  echo $\_GET['name']; //lấy name; |

**\*Lưu ý**:

* Ngoài cách gửi dữ liệu bằng form trên thì vì đây là phương thức GET lên hoàn toàn các bạn có thể truyền tham số bằng tay thông qua paramter trên URL.

Ví dụ :

|  |
| --- |
| http://localhost/get.php?name=lethanhtuan&age=25 |

II.JSON

1. **JSON là gì:**

*JSON là chữ viết tắt của Javascript Object Notation*

Ví dụ dưới đây mình định nghĩa một chuỗi JSON lưu trữ thông tin cá nhân của mình như sau, đây là ví dụ quan trọng nhất của phần json là gì.

|  |
| --- |
| {  "username" : "thanhtuanvietpro",  "email" : "thanhtuanvietpro@gmail.com",  "website" : "vietpro.com",  "title" : "Học lập trình với JSON"  } |

Như vậy cú pháp của JSON rất đơn giản là mỗi thông tin dữ liệu sẽ có 2 phần đó là key và value, điều này tương ứng trong CSDL là tên field và giá trị của nó ở một record nào đó. Tuy nhiên nhìn qua thì đơn giản nhưng nếu ta mổ xẻ nó ra thì có một vài điều như sau:

Chuỗi JSON được bao lại bởi dấu ngoặc nhọn {}

Các key, valuecủa JSON bắt buộc phải đặt trong dấu nháy kép {"}, nếu bạn đặt nó trong dấu nháy đơn thì đây không phải là một chuỗi JSON đúng chuẩn. Nên trường hợp trong value của bạn có chứa dấu nháy kép thì hãy dùng dấu (\) để đặt trước nó nhé, ví dụ học \"json là gì\" tại website vietpro.com.

Nếu có nhiều dữ liệu (nhiều cặp key => value) thì ta dùng dấu phẩy (,) để ngăn cách

Các key của JSON bạn nên đặt chữ cái không dấu hoặc số, dấu \_ và không có khoảng trắng., ký tự đầu tiên không nên đặt là số. Điều này rất giống với nguyên tắc đặt tên biến trong PHP.

Tới đây bạn đã trả lời được cho câu hỏi JSON là gì và tại sao lại sử dụng JSON rồi đấy.

1. **Cấu trúc chuỗi JSON**

Bây giờ chúng ta sẽ làm một vài ví dụ liên quan đến JSON, vì chúng ta chưa sử dụng JSON kết hợp với các ngôn ngữ lập trình nên trong các ví dụ dưới đây chỉ ở mức độ là xây dựng chuỗi JSON đúng chuẩn. Thông qua các ví dụ này các bạn sẽ biết được cấu trúc một chuỗi JSON là gì.

Ví dụ 1: Có 3 sinh viên gồm các thông tin sau: Nguyễn Văn Cường - 21 tuổi, Nguyễn Văn Kính - 22 tuổi, Nguyễn Văn Chính - 23 tuổi. Hãy xây dựng chuỗi JSON lưu trữ thông tin danh sách sinh viên đó.

Nếu xem ví dụ ở phần 1 JSON là gì thì bạn thấy không thể lưu được, vì nếu lưu thì ta sẽ lưu thế này:

|  |
| --- |
| {  "sinhvien1\_ten" : "Nguyễn Văn Cường",  "sinhvien1\_tuoi" : "21 Tuổi",  "sinhvien2\_ten" : "Nguyễn Văn Kính",  "sinhvien2\_tuoi" : "22 Tuổi",  "sinhvien3\_ten" : "Nguyễn Văn Chính",  "sinhvien4\_tuoi" : "22 Tuổi"  } |

Nhìn vào không hay đúng không nào, vì thông thường mỗi đối tượng chúng ta sẽ gom lại một nơi lưu trữ để tiện cho việc tìm kiếm và phân chia, nhưng chuỗi này không lưu được như vậy. Bây giờ ta sẽ tìm hiểu một cách khác nhé.

|  |
| --- |
| [  {  "name" : "Nguyễn Văn Cường",  "age" : "21 tuổi"  },  {  "name" : "Nguyễn Văn Kính",  "age" : "22 tuổi"  },  {  "name" : "Nguyễn Văn Chính",  "age" : "23 tuổi"  }  ] |

Nhìn gọn hơn rồi đúng không nào, chúng ta sử dụng dấu ([]) để gom nhóm lại, và bên trong là danh sách các chuỗi JSON con được cách nhau bởi dấu phẩy (,). Lưu ý là các bạn phải tuân theo những quy tắc JSON mà phần 1 JSON là gì đã trình bày nhé.

Ví dụ 2: Giả sử chúng ta có bài toán như sau, ở một trường đại học lưu trữ điểm của sinh viên, mỗi sinh viên sẽ có một mã số sinh viên sẽ đăng ký học các tín chỉ khác nhau, hãy tổ chức cấu trúc chuỗi JSON để lưu trữ danh sách sinh viên và danh sách các môn học của sinh viên đó.

Giả sử:

* Nguyễn Văn Cường có ID là sv0001
* Nguyễn Văn Kính có ID là sv0002

Bài này hơi khó rồi đấy nhỉ, các bạn tham khảo lời giải sau và rút ra bài học nhé.

|  |
| --- |
| {  "sv0001" : {  "toan" : "Môn Toán",  "ly" : "Môn Lý"  },  "sv0002" : {  "toan" : "Môn Toán",  "anh" : "Môn Anh"  }  } |

Các bạn thấy khác với ví dụ 1 mình không sử dụng dấu [] để bao ngoài cùng mà thay bằng cặp dấu {}, tại sao? Đơn giản là vì cặp [] dùng để bao các chuỗi JSON con, mà chuỗi JSON thì bắt buộc có cặp {} nên trong trường hợp trên không sử dụng được (các cặp con ở dạng key => {value})

1. **Xử lý JSON trong PHP**

PHP cũng là một ngôn ngữ lập trình nên nó cũng có cung cấp các thư viện xử lý JSON giúp lập trình viên giải quyết nó dễ dàng. Chúng ta có hai hàm đó là hàm json\_decode và json\_encode:

**a. Hàm json\_decode trong php**

Hàm này mục đích chuyển một chuỗi JSON sang dạng mảng hoặc object, hàm này có cú pháp như sau: json\_decode($json\_string, $assoc). Trong đó:

* **$json\_string**: là chuỗi JSON
* **$assoc** có hai giá trị true / false. Nếu true thì kết quả nó trả về là dạng array, ngược lại nếu false thì kết quả trả về dạng object. Mặc định là false.

Ví dụ: Cho chuỗi JSON như sau:

|  |
| --- |
| $json\_string =  '  {  "name" : "Lê Thanh Tuấn",  "email" : "thanhtuanvietpro@gmail.com",  "website" : "vietpro.com"  }  '; |

Hãy dùng hàm json\_decode để chuyển về dạng mảng và object. Bài giải như sau:

|  |
| --- |
| $json\_string =  '  {  "name" : "Lê Thanh Tuấn",  "email" : "thanhtuanvietpro@gmail.com",  "website" : "vietpro.com"  }  '; |

// Dạng Mảng

|  |
| --- |
| var\_dump(json\_decode($json\_string, true)); |

// Dạng Object

|  |
| --- |
| var\_dump(json\_decode($json\_string)); |

Kết quả:

|  |
| --- |
|  |

**b. Hàm json\_encode trong PHP**

Hàm này có chức năng ngược lại hàm json\_decode, nó sẽ chuyển một mảng trong PHP hoặc object trong PHP thành chuỗi JSON. Cú pháp như sau: json\_encode($array), trong đó $array là mảng ban muốn chuyển đổi. Kết quả chuỗi JSON sẽ tự động chuyển các ký tự có dấu, các ký tự đặc biệt sang dạng an toàn nên bạn nhìn vào nó hơi khác. Xem ví dụ dưới đây.

Ví dụ: Sử dụng hàm json\_encodeđể chuyển đổi mảng sau sang chuỗi JSON.

|  |
| --- |
| $array = array(  "name" : "Lê Thanh Tuấn",  "email" : "thanhtuanvietpro@gmail.com",  "website" : "vietpro.com"  ); |

Bài giải như sau:

|  |
| --- |
| $array = array(  "name" : "Lê Thanh Tuấn",  "email" : "thanhtuanvietpro@gmail.com",  "website" : "vietpro.com"  );    echo json\_encode($array); |

Kết quả:

|  |
| --- |
| A screenshot of a cell phone  Description automatically generated |

III.Dự án website đa giao diện – đa ngôn ngữ (Việt-Anh)

Yêu cầu:

* Chuyển đổi style trong 1 trang sang các màu tương ứng.
* Chuyển đổi tiếng Anh và tiếng Việt.

A screenshot of a cell phone

Description automatically generated

Hướng dẫn:

* Chuyển style:

Đầu tiên các bạn truyền vào đường dẫn 1 cái biến (bắt đầu từ index.php).

Sau đó mở phân vùng PHP code:

|  |
| --- |
| if(isset($\_GET['…'])){  #code…;  }  else{  #code…;  } |

Rồi echo ra kết quả (làm sao cho cái style nó động tương ứng với mục chúng ta chọn).

* Chuyển ngôn ngữ:

Mở phân vùng php: (Tìm điều kiện isset)

|  |
| --- |
| if(isset($\_GET['…'])){  #code…;  }  else  {  #code…;  }  //lấy dữ liệu từ file json:  $mang = file\_get\_contents($file\_name.'.json');  //chuyển một chuỗi json sang mảng:  $ket\_qua = json\_decode($mang, true); |

Và rồi echo ra kết quả tương ứng.

---/\*Hết\*/---