

Métodos GET e POST no PHP

Prof.: Alisson G. Chiquitto
chiquitto@unipar.br

GET/POST

- Existem 2 maneiras do navegador enviar informações para o servidor web:
 - Método GET;
 - Método POST;

Codificação URL

- Antes do navegador enviar a informação, ele codifica a informação usando um esquema chamado de **codificação de URL**.
 - Os pares nome/valor são unidos com sinal de = (igual) e pares diferentes são unidos pelo & (e comercial);
 - Caracteres não alfanuméricos são substituídos pelo seu valor hexadecimal;
 - Ex.: name1=value1&name2=value2&name3=value3
- Somente após a codificação que a requisição é feita para o servidor;

Método GET

- O método GET envia a informação codificada junto com a URL da página;
- A página e a informação codificada são separados pelo caracter ? (interrogação);
- Ex.: `http://www.test.com/index.php?name1=value1&name2=value2`

Método GET

- Características:
 - Produz uma longa string que aparece na barra de endereço dos navegadores;
 - Restrito a 1024 caracteres;
 - Nunca usar GET para enviar senhas ou informações sigilosas;

GET no PHP

- O PHP fornece uma variável `$_GET` para acessar as as informações enviadas por GET;
- A variável `$_GET` é um array associativo:

```
3  /**
4   * arquivo.php?nome=Unipar&cidade=Cianorte
5   */
6
7  echo $_GET['nome']; // Unipar
8
9  echo $_GET['cidade']; // Cianorte
```

Método POST

- O método POST envia as informações via cabeçalhos HTTP;
- Por POST, não há restrições no tamanho das informações enviadas;
- Como os dados são enviados via cabeçalho HTTP, se for utilizado HTTPS as informações estarão seguras;

POST no PHP

- O PHP fornece uma variável `$_POST` para acessar as as informações enviadas por POST;
- A variável `$_POST` é um array associativo:

POST no PHP

```
<form action="arquivo.php" method="post">  
  Nome: <input type="text" name="nome"><br>  
  Email: <input type="text" name="email"><br>  
  
  <input type="submit" name="enviar" value="Enviar">  
</form>
```

```
<?php
```

```
echo $_POST['nome'];  
echo $_POST['email'];
```

```
?>
```

Função isset()

- A função `isset()` determina se a variável existe e não é NULL;
- <http://php.net/isset>

Função isset()

```
5 // Se $_GET['nome'] existir
6 if (isset($_GET['nome'])) {
7     echo $_GET['nome'];
8 }
9 else {
10     echo 'Nome não existe em GET';
11 }
12
13 // Se $_POST['nome'] existir
14 if (isset($_POST['nome'])) {
15     echo $_POST['nome'];
16 }
17 else {
18     echo 'Nome não existe em POST';
19 }
```

Foreach com GET/POST

- Como GET/POST são arrays associativos, é possível fazer iterações usando foreach();

```
/**
 * arquivo.php?var1=valor1&var2=valor2
 */

foreach($_GET as $k => $v) {
    echo $k . ', ';
}

/**
 * Resultado: var1, var2,
 */
```

Atividade 1

- Criar um programa PHP que:
 - Se `$_GET['n1']` existir, escreve-o;
 - Se `$_GET['n2']` existir, escreve-o;
 - Se `$_GET['n1']` e `$_GET['n2']` existir, escreve a soma entre os dois;

Atividade 2

- Criar um programa que exiba todas as informações vindas pelo método \$_GET. Siga o exemplo da imagem.

```
nome = Arnold  
email = arnold@gmail.com  
idade = 99
```

Atividade 3

- Capture a informação nome disponível via GET, e de uma mensagem de boas vindas para o usuário de acordo com o horário (pesquise a função date()):
 - Exemplo da mensagem se período = matutino:
“Olá Amanda, bom dia”;

Atividade 4

- Criar um formulário HTML que envia as informações por POST. Este formulário deverá conter as seguintes entradas: nome, e-mail, idade, cidade e uf;
- Criar um programa PHP que receba as informações do formulário acima, e mostre apenas as que foram preenchidas pelo usuário;

Referencias

- “PHP GET and POST Method” -
http://www.tutorialspoint.com/php/php_get_post.htm

-