



Explorando Métodos HTTP POST e GET em PHP

Professor: Marco Aurélio Schünke

O que são métodos HTTP?

HTTP (HyperText Transfer Protocol) é o protocolo usado para comunicação entre cliente (navegador) e servidor (onde o site está hospedado).

Quando enviamos dados de um formulário HTML para o servidor usando **PHP**, geralmente utilizamos dois métodos principais:

- **GET** — Dados enviados pela URL.
- **POST** — Dados enviados no corpo da requisição.

Método GET

Características:

- Os dados são enviados anexados à URL, após um ?.
- Ficam visíveis na barra de endereço.
- Usado para requisições rápidas, pesquisa e navegação.
- Limite de caracteres (varia de acordo com o navegador, geralmente até ~2000 caracteres).
- Menos seguro, pois os dados ficam expostos.

Quando usar:

- Busca de informações (ex.: pesquisa no Google).
- Quando não há necessidade de esconder os dados.

Exemplo HTML (GET):

```
<form method="get" action="processa_get.php">  
  Nome: <input type="text" name="nome"><br>  
  Idade: <input type="number" name="idade"><br>  
  <input type="submit" value="Enviar">  
</form>
```



COLÉGIO ESTADUAL PROTÁSIO ALVES

Av. Ipiranga, 1090 – Cep 90160-091 – Porto Alegre – RS

Fones: (51) 3223.2661 e 3223.7066 – Fax: ramal 21

processa_get.php:

```
<?php
// Captura dados enviados pelo método GET
$nome = $_GET['nome'];
$idade = $_GET['idade'];

echo "Nome: " . htmlspecialchars($nome) . "<br>";
echo "Idade: " . htmlspecialchars($idade);
?>
```

Observação: No navegador, após enviar o formulário, a URL ficará assim:

processa_get.php?nome=Joao&idade=25

Método POST

Características:

- Os dados são enviados no corpo da requisição (não aparecem na URL).
- Mais seguro que GET (mas não criptografa por padrão).
- Não há limite prático de tamanho para os dados enviados.

Quando usar:

- Envio de dados sensíveis (ex.: login, senha).
- Envio de formulários grandes.
- Upload de arquivos.

Exemplo HTML (POST):

```
<form method="post" action="processa_post.php">
  Usuário: <input type="text" name="usuario"><br>
  Senha: <input type="password" name="senha"><br>
  <input type="submit" value="Entrar">
</form>
```



processa_post.php:

```
<?php
// Captura dados enviados pelo método POST
$usuario = $_POST['usuario'];
$senha = $_POST['senha'];

// Exibindo (apenas para teste – não fazer isso em sistemas reais!)
echo "Usuário: " . htmlspecialchars($usuario) . "<br>";
echo "Senha: " . htmlspecialchars($senha);
?>
```

Observação: Após enviar, os dados **não aparecem** na URL.

Diferenças principais entre GET e POST

Característica	GET	POST
Visibilidade	Aparece na URL	Oculto na URL
Segurança	Menos seguro	Mais seguro (mas não criptografa)
Limite de tamanho	Sim (~2000 chars)	Não significativo
Uso recomendado	Consultas, busca	Login, cadastros, uploads
Cache do navegador	Pode ser cacheado	Não cacheado

Boas práticas

- **Validar e sanitizar dados** antes de usar (`htmlspecialchars`, `filter_input`, etc.).
 - Usar **POST** para dados sensíveis.
 - Evitar expor informações privadas no método GET.
 - Usar **HTTPS** para proteger a transmissão de dados.
-



COLÉGIO ESTADUAL PROTÁSIO ALVES

Av. Ipiranga, 1090 – Cep 90160-091 – Porto Alegre – RS

Fones: (51) 3223.2661 e 3223.7066 – Fax: ramal 21

Exemplo final — mesmo formulário com GET e POST

formulario.php:

```
<h2>Formulário com GET</h2>
<form method="get" action="processa.php">
  Nome: <input type="text" name="nome"><br>
  <input type="submit" value="Enviar GET">
</form>

<h2>Formulário com POST</h2>
<form method="post" action="processa.php">
  Nome: <input type="text" name="nome"><br>
  <input type="submit" value="Enviar POST">
</form>
```

processa.php:

```
<?php
if ($_SERVER['REQUEST_METHOD'] == 'GET') {
    echo "Recebido via GET: " . htmlspecialchars($_GET['nome']);
}
elseif ($_SERVER['REQUEST_METHOD'] == 'POST') {
    echo "Recebido via POST: " . htmlspecialchars($_POST['nome']);
}
?>
```