

**IFBA EUCLIDES DA CUNHA**

**2024**

**ADERALDO SANTANA CAMPOS JUNIOR  
PEDRO ANDRADE SANTANA  
AILSON CRUZ SOARES**

**MÉTODOS GET E POST EM PHP**

**EUCLIDES DA CUNHA 2024**

## **O que são os métodos POST e GET**

Tanto GET como POST na verdade são métodos HTTP. Eles indicam para o servidor qual a ação que o cliente deseja realizar. Quando realizamos uma requisição obrigatoriamente precisamos informar um método.

GET – é usado quando o cliente deseja obter recursos do servidor

POST – é usado quando o cliente deseja enviar dados para processamento ao servidor, como os dados de um formulário, por exemplo.

Existem outros métodos HTTP. Os dois métodos citados acima são os mais usados, principalmente em aplicações web. Quando o usuário digita um endereço e aperta enter na barra de endereço do navegador, ele realiza uma requisição do tipo GET. Já quando preenchemos um formulário e clicamos em enviar geralmente o método usado é o POST.

## **Qual a diferença do método POST e GET?**

O método GET é usado para solicitar dados de um recurso específico no servidor. Quando um formulário é enviado usando o método GET, os dados do formulário são anexados à URL como uma string de consulta (query string). Isso significa que os dados são visíveis na barra de endereços do navegador. O método GET é adequado para solicitações onde os dados não são sensíveis e não modificam o estado do servidor. É comumente usado para pesquisas, filtros e para navegar entre diferentes páginas de um site.

No método GET:

- Os dados do formulário são enviados como parte da URL.
- Os dados são visíveis na barra de endereços do navegador.
- É adequado para formulários que enviam pequenas quantidades de dados, como pesquisas ou filtros.
- Limitado em tamanho, geralmente limitado a cerca de 2048 caracteres.
- Pode ser cacheado pelo navegador, o que pode representar um risco de segurança se os dados enviados forem confidenciais.
- Não deve ser usado para enviar dados sensíveis, como senhas, pois os dados são visíveis na URL.

O método POST é usado para enviar dados para serem processados por um script no servidor, sem que esses dados sejam visíveis na URL. Ao contrário do método GET, os dados enviados pelo método POST são incluídos no corpo da solicitação HTTP, não na URL.

O POST é mais adequado para situações onde os dados são sensíveis, como informações de login, submissão de formulários com dados privados, ou quando os dados enviados são muito grandes para serem incluídos na URL. Ele também é mais seguro em termos de confidencialidade, pois os dados não são expostos na barra de endereços do navegador e não são armazenados no histórico de navegação.

No método POST, os dados do formulário são enviados no corpo da solicitação HTTP.

- Os dados não são visíveis na barra de endereços do navegador.
- Pode ser usado para enviar grandes quantidades de dados, como uploads de arquivos.
- Não tem limitação de tamanho.
- Não é cacheado pelo navegador.
- Mais seguro para enviar dados sensíveis, pois os dados não são visíveis na URL.

Em resumo a diferença principal entre os dois métodos, é basicamente, o envio de informações para o servidor. Onde método GET, transfere os dados para o serviço através da URL, e a quantidade de caracteres é limitado, Já o método POST, você pode realizar envios mais seguros, e não há limite para transmissão de informações.

## DESVANTAGENS

Embora os métodos GET e POST sejam amplamente usados em desenvolvimento web, cada um tem suas próprias desvantagens. Aqui estão algumas desvantagens associadas a cada método em PHP:

### Desvantagens do método GET:

1. **Limitação de Tamanho:** Os dados enviados via método GET são anexados à URL, o que pode levar à limitação do tamanho dos dados transmitidos. Os navegadores e servidores podem impor um limite na quantidade de dados que podem ser incluídos em uma URL. Isso pode ser problemático ao tentar enviar grandes volumes de dados, resultando em truncamento ou perda de informações.
2. **Segurança:** Como os dados são enviados como parte da URL, eles são visíveis para o usuário e podem ser facilmente interceptados e modificados. Isso pode representar um risco de segurança, especialmente ao lidar com informações sensíveis, como senhas ou detalhes financeiros. Além disso, os dados enviados via método GET são armazenados no histórico do navegador e nos logs do servidor, o que pode expor informações confidenciais.
3. **Cache do Navegador:** Os resultados de solicitações GET podem ser armazenados em cache pelo navegador ou por proxies, o que pode resultar em problemas de cache e exibir

informações desatualizadas para o usuário. Isso pode ser especialmente problemático em casos onde os dados são dinâmicos e estão sujeitos a alterações frequentes.

### **Desvantagens do método POST:**

1. **Segurança Limitada:** Embora o método POST seja mais seguro do que o GET, os dados ainda podem ser interceptados e manipulados, embora de forma menos direta. A segurança depende muito da implementação correta de práticas de segurança, como a validação de entrada e a prevenção de ataques de injeção.
2. **Complexidade da Implementação:** O método POST pode ser mais complexo de implementar do que o GET, especialmente ao lidar com formulários HTML complexos ou ao processar uploads de arquivos. A necessidade de configurar corretamente os cabeçalhos HTTP e o manuseio do fluxo de dados podem aumentar a complexidade do código.
3. **Uso Ineficiente de Recursos:** Como os dados são enviados no corpo da solicitação HTTP, o método POST pode consumir mais recursos do servidor, especialmente ao lidar com grandes volumes de dados. Isso pode resultar em um aumento do uso de largura de banda e tempo de processamento, especialmente em sistemas com alta carga de tráfego.

Em resumo, embora tanto o método GET quanto o POST sejam úteis em diferentes situações, é importante entender suas limitações e considerar os requisitos específicos do seu projeto ao escolher o método mais adequado para o envio de dados em PHP.

## **Qual escolher?**

Para a utilização do nosso projeto, é mais viável optar pelo método POST, pois ele é mais recomendado para enviar dados sensíveis e complexos do cliente para o servidor. Isso ocorre porque o método POST envia os dados no corpo da solicitação HTTP, em vez de anexá-los à URL como no método GET.

Ao escolher o método POST, garantimos as seguintes vantagens adequadas ao projeto:

**Segurança dos dados:** os dados enviados via POST não são visíveis na barra de endereços do navegador, o que os torna mais seguros.

**Limite de tamanho dos dados:** o método POST é mais adequado, pois não tem esse limite.

**Persistência dos dados:** Usar o método POST ajuda a garantir que os dados enviados sejam mais confiáveis, pois não são facilmente manipulados pelo usuário.

**Conformidade com padrões de segurança:** O do método POST para transmitir informações sensíveis de forma segura.