

Curso: Análise e Desenvolvimento de Sistemas

Disciplina: Linguagem de Programação IV (PHP)

Cookies – usos

1. INTRODUÇÃO

Cookies são recursos para armazenar informações no front-end da aplicação. A Operação de preservação dos dados fica por conta do navegador que está em uso.

Segundo o manual do php.

“O PHP suporta de forma transparente cookies HTTP. Cookies são um mecanismo para armazenar dados no navegador remoto e assim rastrear ou identificar usuários que retornam. Você pode criar cookies usando as funções `setcookie()` ou `setrawcookie()`. Cookies fazem parte do cabeçalho HTTP, logo, `setcookie()` deve ser chamada antes que qualquer saída seja enviada ao navegador. Esta é a mesma limitação que a função `header()` tem. Você pode usar as funções de buffering de saída para atrasar as impressões no script até que você tenha decidido se vai ou não configurar qualquer cookie ou enviar quaisquer cabeçalhos.

Qualquer cookie enviado para o servidor pelo cliente será automaticamente incluído no array auto-global `$_COOKIE` se `variables_order` contém "C". Se você deseja definir vários valores em um único cookie, simplesmente acrescente `[]` ao nome do cookie.

Em sistemas PHP mais antigos (5.3 ou anterior), a diretiva `register_globals`, pode estar ativada, o que pode causar operação indesejável e insegura. Se esta estiver ativada, os cookies serão registrados como variáveis globais.” (PHP - Cookies)

Vale notar que o padrão de instalação do PHP no ambiente XAMP determina o parâmetro `variables_order` com o conteúdo "GPCS", ou seja, as variáveis superglobais `$_GET`, `$_POST`, `$_COOKIE`, `$_SESSION` sempre serão criadas.

A informação armazenada no Cookie tem um tempo de vida, após este tempo o cookie torna-se inválido.

Diferente da informação armazenada em Session, um cookie é armazenada localmente no computador, logo computadores diferentes possuirão informações diferentes, pois não há um ponto central de armazenamento, assim um site visitado em um computador A, não poderá reconhecer automaticamente o usuário se for utilizado o computador B em outro momento.

2. ARMAZENAR INFORMAÇÃO EM COOKIE.

O armazenamento de um cookie é feito utilizando o a função `setcookie()`, abaixo segue o detalhamento da função e seus parâmetros de funcionamento segundo o manual do PHP.

Curso: Análise e Desenvolvimento de Sistemas

Disciplina: Linguagem de Programação IV (PHP)

“setcookie — Envia um cookie

Descrição

setcookie (string \$name [, string \$value = "" [, int \$expire = 0 [, string \$path = "" [, string \$domain = "" [, bool \$secure = false [, bool \$httponly = false]]]]]) : bool

A função setcookie() define um cookie para ser enviado juntamente com o resto dos cabeçalhos HTTP. Como outros cabeçalhos (headers), os cookies devem ser enviados antes de qualquer saída do seu script (isso é uma restrição do protocolo). O que quer dizer que você deve colocar chamadas a essa função antes de qualquer saída, incluindo as tags <html> e <head> e também espaços em branco.

Uma vez que os cookies foram setados, eles podem ser acessados no próximo carregamento da página através do array \$_COOKIE. Os valores dos cookies também podem existir no \$_REQUEST.

Parâmetros ¶

A RFC 6265 fornece a referência normativa de como cada parâmetro de setcookie() é interpretado.

name

O nome do cookie.

value

O valor do cookie. Esse valor é guardado no computador do cliente; não guarde informação sensível. Supondo que o name seja 'nomedocookie', o valor pode ser lido através de \$_COOKIE['nomedocookie']

expire

*O tempo para o cookie expirar. Esse valor é uma timestamp Unix, portanto é o número de segundos desde a época (epoch). Em outras palavras, você provavelmente irá utilizar isso com a função time() mais o número de segundos que você quer que ele expire. Ou você pode utilizar a função mktime(). time()+60*60*24*30 irá configurar o cookie para expirar em 30 dias. Se configurado para 0, ou omitido, o cookie irá expirar ao fim da sessão (quando o navegador fechar).*

Nota:

Você pode ver que o parâmetro expire recebe uma timestamp Unix, ao contrário do formato de data Wdy, DD-Mon-YYYY HH:MM:SS GMT, isso se dá porque o PHP faz essa conversão internamente.

path

O caminho no servidor aonde o cookie estará disponível. Se configurado para '/', o cookie estará disponível para todo o domain. Se configurado para o diretório '/foo/', o cookie estará disponível apenas dentro do diretório /foo/ e todos os subdiretórios como /foo/bar do domain. O valor padrão é o diretório atual onde o cookie está sendo configurado.

domain

O (sub)domínio para qual o cookie estará disponível. Definindo para um subdomínio (como 'www.example.com') deixará o cookie disponível para aquele subdomínio e todos os outros sub-domínios abaixo dele (exemplo w2.www.example.com). Para deixar o cookie

Curso: Análise e Desenvolvimento de Sistemas

Disciplina: Linguagem de Programação IV (PHP)

disponível para todo o domínio (incluindo todos os subdomínios dele), simplesmente defina o valor para o nome do domínio ('example.com', nesse caso).

Browsers antigos ainda implementam a » RFC 2109 e podem requerer um . no início para funcionar com todos os subdomínios.

secure

Indica que o cookie só poderá ser transmitido sob uma conexão segura HTTPS do cliente. Quando configurado para TRUE, o cookie será enviado somente se uma conexão segura existir. No lado do servidor, fica por conta do programador enviar esse tipo de cookie somente sob uma conexão segura (ex respeitando \$_SERVER["HTTPS"]).

httponly

Quando for TRUE o cookie será acessível somente sob o protocolo HTTP. Isso significa que o cookie não será acessível por linguagens de script, como JavaScript. É dito que essa configuração pode ajudar a reduzir ou identificar roubos de identidade através de ataques do tipo XSS (entretanto ela não é suportada por todos os browsers), mas essa informação é constantemente discutida. Foi adicionada no PHP 5.2.0. TRUE ou FALSE

Valor Retornado ¶

Se existe saída antes da chamada dessa função, setcookie() irá falhar e retornará FALSE. Se a função setcookie() for executada com sucesso, ela retornará TRUE. Isso não indica que o usuário aceitou o cookie.” (PHP - setcookie())

Sendo assim, na intenção de armazenar uma informação em um cookie por 12 horas, será necessário estabelecer o nome do cookie, a informação que será armazenada e o tempo para ele expirar.

Consideremos as seguintes informações:

Nome do cookie : id_usuario;

Informação: “LP2020”;

Tempo para expirar: time() + (60*60*12), onde os time() é uma função do PHP que retorna a data e hora atual e os valores fixo 60*60*12, significam respectivamente, 60 segundos, 60 minutos e 12 horas, pois, o tempo para expirar um cookie é determinado em segundos.

A linha de comando para o exemplo acima é:

```
setcookie("id_usuario", "LP2020", time()+(60*60*12));
```

Para a situação onde a informação que será guardada estiver em uma variável, será utilizado no lugar do texto da informação o nome da variável.

Curso: Análise e Desenvolvimento de Sistemas

Disciplina: Linguagem de Programação IV (PHP)

Ex: com a variável \$id.

```
$id = "LP2020";  
setcolor("id_usuario", $id, time()+(60*60*12));
```

Se o objetivo fosse manter o cookie válido por dois dias, seriam utilizados os valores 60*60*24*2, 60 segundo, 60 minutos, 24 horas e 2 dias.

É importante ressaltar que diferentemente de dados gravados em Session, o cookie perderá sua validade no tempo estipulado, mesmo que você continue acessando as páginas, caso não exista uma atualização do tempo de expiração do cookie.

Um exemplo desta rotina pode ser analisada no projeto Uso_Cookies, disponibilizado no moodle

3. ARMAZENAR VETORES EM COOKIE

O cookie foi desenvolvido para armazenar conteúdos em texto (string), por isto, vetores (array) não podem ser armazenados diretamente.

Uma forma de armazenar vetores em um cookie é transformando o vetor em uma cadeia de caracteres, e para isto é utilizado a função serialize().

A função unserialize() realizada a operação inversa, transformando uma cadeia de caracteres em um vetor. Isto se um vetor foi utilizado na transformação da função serialize().

Abaixo está o texto explicativo da função serialize() e unserialize().

“Serialização de Objetos - objetos em sessão

A função serialize() retorna uma string contendo uma representação byte-stream de qualquer valor que pode ser armazenado pelo PHP. A função unserialize() pode utilizar essa string para recriar os valores originais da variável. Utilizar a serialização para salvar um objeto, salvará todas as variáveis de um objeto. Os métodos de um objeto não serão salvos, apenas o nome da classe.

Para deserializar um objeto com a função unserialize(), a classe deste objeto precisar estar definida. Isto é, se possuir um objeto da classe A e serializá-lo, você obterá uma string que referencia a classe A e contém todos os valores de variáveis contidas nele. Se deseja-se deserializar este em outro arquivo, um objeto da classe A, a definição da classe A deve estar presente neste arquivo. Isso pode ser feito, por exemplo, armazenando a definição da classe A em um arquivo, e incluí-lo, fazendo uso da função spl_autoload_register().”
(PHP - Serialização)

serialize

(PHP 4, PHP 5, PHP 7)

Curso: Análise e Desenvolvimento de Sistemas

Disciplina: Linguagem de Programação IV (PHP)

“serialize - Gera uma representação armazenável de um valor

Descrição

serialize (mixed \$value) : string

Gera uma representação armazenável de um valor.

Isso é útil para armazenar ou transmitir valores PHP sem perder seu tipo e estrutura.

Para transformar a string serializada em um valor PHP novamente, use unserialize ().

Parâmetros ¶

valor

O valor a ser serializado. serialize () lida com todos os tipos, exceto o tipo de recurso e alguns objetos (veja a nota abaixo). Você pode até serializar () matrizes que contêm referências a si mesmas. As referências circulares dentro da matriz / objeto que você está serializando também serão armazenadas. Qualquer outra referência será perdida.

Ao serializar objetos, o PHP tentará chamar a função membro __sleep () antes da serialização. Isso permite que o objeto faça qualquer limpeza de última hora, etc. antes de ser serializado. Da mesma forma, quando o objeto é restaurado usando unserialize (), a função de membro __wakeup () é chamada.

Valor Retornado ¶

Retorna uma string contendo uma representação de valor de fluxo de bytes que pode ser armazenada em qualquer lugar.

Observe que essa é uma sequência binária que pode incluir bytes nulos e precisa ser armazenada e manipulada como tal. Por exemplo, a saída serialize () geralmente deve ser armazenada em um campo BLOB em um banco de dados, em vez de um campo CHAR ou TEXT.” (PHP - Serialize()) – Tradução nossa.

“unserialize - Cria um valor PHP a partir de uma representação armazenada

Descrição ¶

unserialize (string \$str [, array \$options]): mixed

unserialize () pega uma única variável serializada e a converte novamente em um valor PHP.

Aviso

Não transmita entrada de usuário não confiável para unserialize (), independentemente do valor de opções de allowed_classes. A desserialização pode resultar no carregamento e na execução de códigos devido à instanciação e carregamento automático de objetos, e um usuário mal-intencionado pode explorar isso. Use um formato seguro e padrão de intercâmbio de dados, como JSON (via json_decode () e json_encode ()), se precisar passar dados serializados para o usuário.

Curso: Análise e Desenvolvimento de Sistemas

Disciplina: Linguagem de Programação IV (PHP)

Se você precisar desserializar dados serializados armazenados externamente, considere usar `hash_hmac()` para validação de dados. Certifique-se de que os dados não sejam modificados por ninguém além de você.

Parâmetros ¶

str

A cadeia de caracteres serializada.

Se a variável que está sendo desserializada for um objeto, após a reconstrução bem-sucedida do objeto, o PHP tentará automaticamente chamar a função de membro `__wakeup()` (se existir).

Nota: diretiva `unserialize_callback_func`

É possível definir uma função de retorno de chamada que será chamada, se uma classe indefinida for instanciada durante a desserialização. (para evitar a obtenção de um objeto incompleto `__PHP_Incomplete_Class`.) Use seu `php.ini`, `ini_set()` ou `.htaccess` para definir `unserialize_callback_func`. Sempre que uma classe indefinida for instanciada, ela será chamada. Para desativar esse recurso, basta esvaziar essa configuração.

opções

Quaisquer opções a serem fornecidas para desserializar (), como uma matriz associativa.

Opções válidas

Nome Tipo Descrição

`allowed_classes mixed` Um conjunto de nomes de classes que devem ser aceitos, `FALSE` para não aceitar classes ou `TRUE` para aceitar todas as classes. Se essa opção for definida e `unserialize()` encontrar um objeto de uma classe que não deve ser aceito, o objeto será instanciado como `__PHP_Incomplete_Class`. Omitir esta opção é o mesmo que defini-la como `TRUE`: O PHP tentará instanciar objetos de qualquer classe.

Valor Retornado ¶

O valor convertido é retornado e pode ser um valor booleano, inteiro, flutuante, sequência, matriz ou objeto.

Caso a sequência passada não seja desserializável, `FALSE` será retornado e `E_NOTICE` será emitido.” (PHP - `unserialize()`) – Tradução nossa.

4. REFERÊNCIAS

PHP - Cookies Cookies [Online] // PHP.Net. - 19 de 03 de 2020. -

https://www.php.net/manual/pt_BR/features.cookies.php.

PHP - Serialização Serialização de Objetos [Online] // Manual PHP. - 20 de março de 2020. -

https://www.php.net/manual/pt_BR/language.oop5.serialization.php.

PHP - Serialize() Manual do PHP [Online] // PHP. - 20 de março de 2020. - serialize.

PHP - setcookie() Manual do PHP [Online] // Manual do PHP. - 19 de março de 2020. -

https://www.php.net/manual/pt_BR/function.setcookie.php.

Curso: Análise e Desenvolvimento de Sistemas

Disciplina: Linguagem de Programação IV (PHP)

PHP - unserialize() Manual do PHP [Online] // PHP. - 20 de março de 2020. -
https://www.php.net/manual/pt_BR/function.unserialize.php.
