Metodi di invio dei dati (GET e POST)

26. 11. 2019

Jan Barášek

Obsah článku

Oltre alle variabili normali, in PHP esistono anche le cosiddette **variabili superglobali**, che contengono informazioni sulla pagina correntemente chiamata e sui dati passati.
In genere, in una pagina abbiamo un modulo che l'utente compila e vogliamo trasferire i dati al server web, dove li elaboriamo in PHP.

I metodi più comunemente utilizzati per farlo sono tre:

Související články

V jiných jazycích

- POST ~ i dati vengono passati di nascosto insieme alla richiesta di pagina
- Ajax POST ~ elaborazione javascript asincrona

Metodo GET - `$_GET

I dati inviati con il metodo GET sono visibili nell'URL (come parametri dopo il punto interrogativo); la lunghezza massima è di 1024 caratteri in Internet Explorer (gli altri browser *quasi* non lo limitano, ma i testi più grandi non dovrebbero essere passati in questo modo). Il vantaggio di questo metodo è soprattutto la semplicità (si può vedere cosa si sta inviando) e la possibilità di fornire un link al risultato dell'elaborazione. I dati vengono inviati a una variabile.

L'indirizzo della pagina ricevente potrebbe essere il seguente:

```
https://____________.com/script.php?promenna=obsah&promenna2=obsah
```

In PHP, ad esempio, possiamo scrivere il valore del parametro `variabile` come segue:

```
echo $_GET['passeggiata'];  // stampa "contenuto"
```

> Questo metodo di scrittura dei dati direttamente nella pagina HTML non è sicuro, perché si può passare, ad esempio, del codice HTML nell'URL che

4. Slovensky
5. Italiano
6. Polska
7. Deutsch
8. Svenska
9. Español
10. 中国
11. 日本
12. український
13. Dansk

/ PHP Manual

> Dobbiamo **sempre** trattare i dati prima di qualsiasi output alla pagina; la funzione `htmlspecialchars()` è usata per questo.
>
> Per esempio: `echo htmlspecialchars($_GET['variabile']);`

Metodo POST - `$_POST

I dati inviati con il metodo POST non sono visibili nell'URL, il che risolve il problema della lunghezza massima dei dati inviati. Per l'invio dei campi dei moduli si dovrebbe sempre utilizzare il metodo POST, che garantisce che, ad esempio, le password non siano visibili e che non si possa fornire un link alla pagina che elabora il risultato di un determinato input.

I dati sono disponibili nella variabile `$_POST` e l'uso è lo stesso del metodo GET.

Verifica dell'esistenza dei dati inviati

Prima di elaborare qualsiasi dato, è necessario verificare che i dati siano stati effettivamente inviati, altrimenti si accederebbe a un'area di lavoro di un'azienda. a una variabile inesistente, con conseguente messaggio di errore.

La funzione `isset()` viene utilizzata per verificare l'esistenza di una variabile.

/ PHP Manual

```
    echo 'Il tuo nome:' . htmlspecialchars($_GET['Nome']);
} else {
    echo 'Non è stato inserito alcun nome.';
}
```

Modulo di inserimento dati

Il modulo è realizzato in HTML, non in PHP. Può trovarsi in una semplice pagina HTML. Tutta la "magia" è gestita dallo script PHP che accetta i dati.

Ad esempio, possiamo utilizzare un modulo per ricevere 2 numeri, inviati con il metodo GET:

```
<form action="script.php" method="get">
    První číslo: <input type="text" name="x">
    Druhé číslo: <input type="text" name="y">

    <input type="submit" value="Sečíst čísla">
</form>
```

Nella prima riga si può vedere dove verranno inviati i dati e con quale metodo.

Seguono il pulsante per l'invio dei dati (obbligatorio) e il tag HTML di chiusura del modulo (obbligatorio affinché il browser sappia cos'altro inviare e cosa no).

> Possiamo avere un numero qualsiasi di moduli in una pagina e non possono essere annidati. In caso di annidamento, viene sempre inviato il modulo più annidato e gli altri vengono ignorati.

Elaborazione dei moduli sul server

Ora abbiamo un modulo HTML finito e lo inviamo a `script.php` , che riceve i dati con il metodo GET. L'indirizzo della richiesta di pagina potrebbe essere simile a questo:

```
https://________.com/script.php?x=5&y=3
```

script.php

```
$x = $_GET['x'];    // 5
$y = $_GET['y'];    // 3

echo $x + $y;       // stampa 8
```

/ PHP Manual

```
if (isset($_GET['x']) && isset($_GET['y'])) {
    $x = $_GET['x'];    // 5
    $y = $_GET['y'];    // 3

    echo $x + $y;       // stampa 8
} else {
    echo 'Il modulo non è stato compilato correttamente.';
}
```

> **TIP:** È possibile passare più parametri al costrutto `isset()` per verificare che esistano tutti.
>
> Pertanto, invece di `isset($_GET['x']) && isset($_GET['y'])` , si può semplicemente specificare:
>
> `isset($_GET['x'], $_GET['y'])` .

Elaborazione dei dati ricevuti con il metodo POST

PHP Manual

```
https://________.com/script.php
```

E mai altrimenti. Semplicemente no. I dati sono nascosti nella richiesta HTTP e non possiamo vederli.

> Il metodo hidden POST è necessario per inviare nomi utente e password per motivi di sicurezza.
>
> Se sul sito si utilizzano password, il modulo di login e di registrazione deve essere ospitato su HTTPS e le password devono essere sottoposte a un hash appropriato (ad esempio, con BCrypt).

Gestione delle richieste ajax

In alcuni casi, durante l'elaborazione delle richieste ajax, potrebbe non essere facile recuperare i dati. Questo perché le librerie ajax di solito inviano i dati come `json payload`, mentre la variabile superglobale `$_POST` contiene solo i dati del modulo.

I dati sono comunque accessibili; ho descritto i dettagli nell'articolo Elaborazione delle richieste ajax POST.